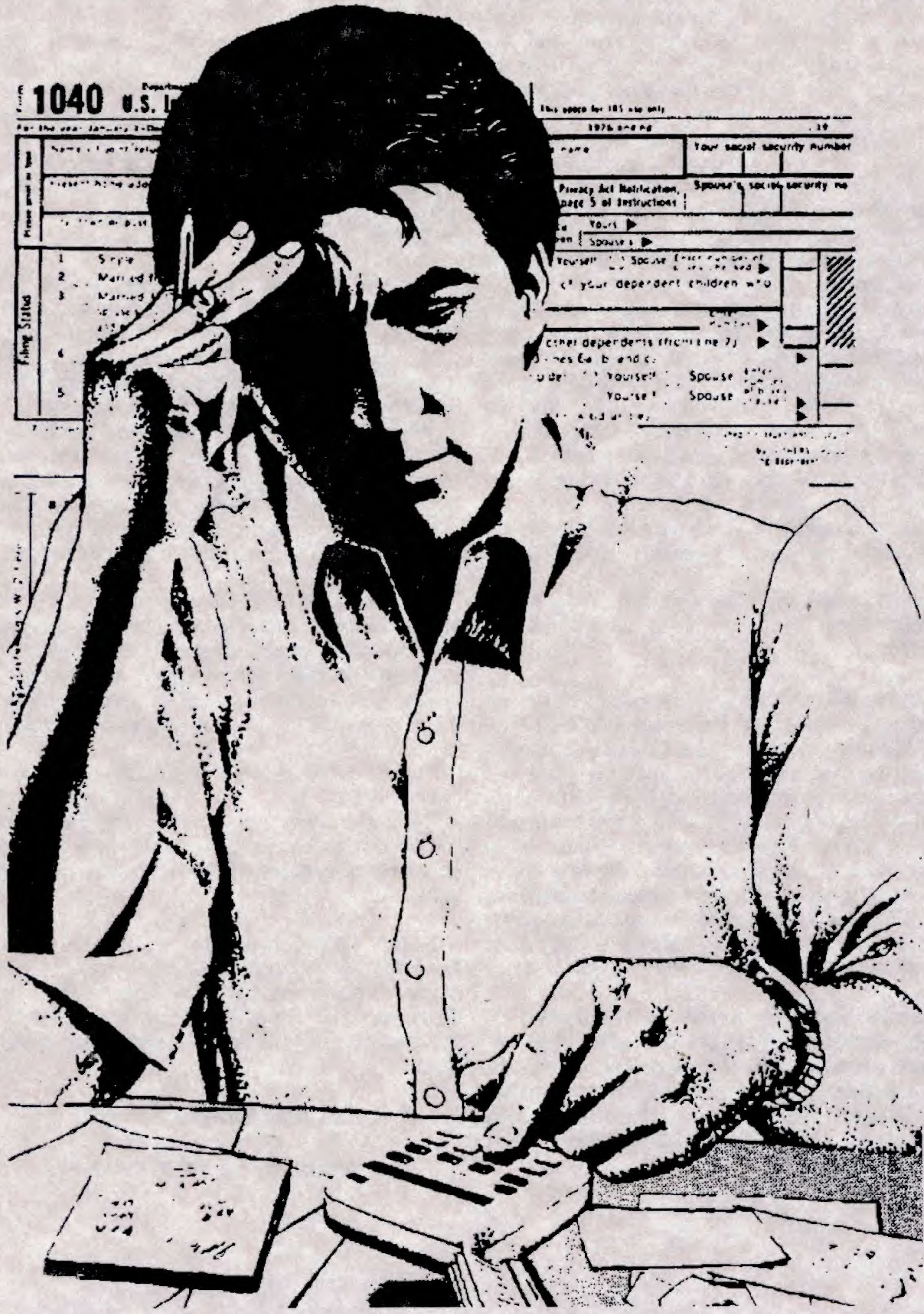
TRSTimes

Volume 5. No. 3. - May/Jun 1992 - $4.00



LITTLE ORPHAN EIGHTY

Though it was a few months late, I finally carried out my New Years resolution for 1992 - I quit smoking!!!

Yes, after decades of puffing away 1-2 packs a day, I finally saw the light. Or was it the cough, along with the constant nagging from my wife and kids, that did the trick?

Anyway, the long and the short of it is that I quit on February 20, 1992. It was just before midnight and I was sitting at the Model 4, writing a future article for TRSTimes, when I reached in my shirt pocket for a cigarette. DRAT - the pack was empty. I crumpled it up and threw it in the wastepaper basket.

I was about to go out to the nearest convenience store for replenishment, when I began one of those deep, throaty coughs that seems to originate from the stomach, and then work its way up, shaking the whole body as if in convulsions. I finally got the message. The heavens opened up and a voice boomed out: "Don't do that no more'". So, not being one to argue with omens, I quit. Instead of going to the store, I shut off the Model 4, took a long shower and went to bed. As of this writing, my formerly nicotine-stained fingers have not touched a cigarette for exactly 2 months. It's getting easier -- they tell me!!

This month's front page reflects the other kind of agony I went through in April. However, I am sure that in this case I was not alone.

Because of a job promotion, a very good friend has left the greater Los Angeles area. Previously a devoted TRS-80 fan, he is currently involved in the IBM side of computing, so he offered his collection of TRS-80 hardware and software for sale before the move. Roy Beck and I are now the proud owners of what is probably one of the largest TRS-80 public domain libraries anywhere. As soon as we sort out what is what, we will make this available on a disk by disk basis, at a reasonable price, through the pages of TRSTimes. Roy, I am sure, will make a favorable deal for one or more of the hard drives. There are many other goodies, we just haven't had a chance to go through it all yet.

For example, I picked up a great Model 4. It has 256K RAM, amber screen, RS hi-res board, the IBM character set chip, and 4 internal drives (two double-sided 40's, one 5.25 inch double-sided 80, and one - 3,5 inch double-sided 80). This is a great machine, and I am really having fun using it. I haven't yet used the extra memory (I can't find the driver).

Well, worse come to worse, I'll write my own. This should prove to be an interesting experience!!

I really do like working with the 3.5 inch disks. They are sturdy, they fit in my shirt pocket and, according to Jim King, the very best part is that there are no sleeves to misplace.

In an earlier issue, this column mentioned that Roy Soltoff was thinking about putting the Misosys Quarterly to rest. The latest word is that he WILL continue to publish. This is good news, so be sure to get your subscription to this essential publication.

The other news from Misosys is that the combined Model I/III & 4 LDOS/LS-DOS manual is now available. I am ordering my copy as soon as this issue of TRSTimes is finished.

I would like to thank Henry Herrdegen and Chris Fara (of Microdex) for taking the time to review DR. PATCH, and for saying nice things about my programming effort.

Wouldn't you know, Henry discovered a minor, but irritating bug. Henry did what my beta-testers failed to do; he purposely tried to make the program crash.

While DR. PATCH did not crash, it did behave strangely if it was told to work on a non-DOS disk. It should simply have reported that particular files were not found, and then returned to the menu; instead, an incorrect error message is displayed, and the program then returns to the menu. However, in the process empty files are written to the target disk!

Though this was not a major goof, it was, nevertheless, a goof. I wrote the routine to check for existing files the way I would in assembler; that is, OPEN the file and branch to an error handler if the file doesn't exist. I forgot - Basic doesn't work that way. If you OPEN a file that doesn't exist, Basic CREATES it. Thus, an empty file is written to the target disk.

Thanks Henry, because of you, DR. PATCH is now up to version 1.2.

Owners of version 1.1 should send the original DR. PATCH disk to TRSTimes; we will send the upgrade by return mail. Ah yes, the trials and tribulations of a software author!!!

Before bringing this column to a close, I would like to thank the other important people whose contributions made this issue possible - Roy Beck, Jack Nock, Mike Ecker, Jim King, Frank Gottschalk, M.C. Matthews, Karl Mohr, and Allen Jacobs - with this kind of help, being editor is easy.

And now, on to the business at hand.....

Welcome to TRSTimes 5.3

TRSTimes magazine

Volume 5. No. 3. - May/Jun 1992

PUBLISHER-EDITOR
Lance Wolstrup
CONTRIBUTING EDITORS
Roy Beck
Dr. Allen Jacobs
Dr. Michael W. Ecker
TECHNICAL ASSISTANCE
San Gabriel Tandy Users Group
Valley TRS-80 Users Group
Valley Hackers' TRS-80 Users Group

TRSTimes magazine is published bi-monthly by TRSTimes Publications. 5721 Topanga Canyon Blvd, Suite 4. Woodland Hills, CA. 91364. (818) 716-7154.
Publication months are January, March, May, July, September and November.

1992 subscription rates (6 issues):
UNITED STATES & CANADA: $20.00 (U.S. currency)
EUROPE, CENTRAL & SOUTH AMERICA: $24.00 for surface mail or $31.00 for air mail. (U.S. currency only)
ASIA, AUSTRALIA & NEW ZEALAND: $26.00 for surface maile or $34.00 for air mail. (U.S. currency only)
Article submissions from our readers are welcomed and encouraged. Anything pertaining to the TRS-80 will be evaluated for possible publication. Please send hardcopy and, if at all possible, a disk with the material saved in ASCII format. Any disk format is acceptable, but please note on label which format is used.

THE MAIL ROOM

BOOT-UP SCREEN

If you are sitting around your office and developing another case of "writer's block", how about coming up with a nice boot screen for LS-DOS 6.3.1!!

Mickey Mepham
Charles City, VA

Someone did develop a very nice screen for LS-DOS 6.3.0. Unfortunately, it doesn't work with version 6.3.1. When the dust settles around here, I will take a look and see if I can convert it to work with 6.3.1.

Ed.

DR. PATCH

The mailman dropped off my DR. PATCH disk earlier today, and I thought I'd let you know how pleased I am with the program. It is easy to use and most of the patches are of great help to me; especially the password removers and the 'KILL' command. I also have fun making different DOS prompts. Keep up the good work.

Duane T. Walker
Anderson, IN

Thank you, Duane. Your nice words are much appreciated.

Ed.

MORE ON MODEMS

This letter is in reaction to an article by Paul Abernathy in the January/February issue about hooking up a modem. It was OK as far as it went, but for the most part I think the article stopped short in terms of what the would-be modem user really needs to know to get started, and why he would want to.

Abernathy concluded that all the modem owner needs to do after hooking it up is choose his favorite software. But how can he know what his favorite software is, and how does know what he wants to do with it?

What follows almost could be an article offered for publication rather than a letter, except my knowledge of all the communication packages available for the Model 4 is limited. I can't compare them; all I can tell you are my experiences.

I was put off from using a modem for years by the intimidating comm packages that came bundled with TRSDOS, LDOS and LS-DOS, and it wasn't until I discovered XT4, available on public domain disks, and its ability to interface with the Hayes command set of today's modems, that communications became something I could understand. Numbers to call are all in a directory that keeps track of technical settings and passwords, so that I don't have to. XT4 dials the numbers for me. The tap of one key opens RAM's capture buffer to collect text coming to me, or instructs the printer to copy everything that comes onto the screen, or permits uploadinug or downloading of a disk file.

So what do I do with a modem, which I've had for only a few months although our household has had a TRS-80 since 1985.

First, the family joined GEnie, which for $4.95 a month is a treasure trove of information and consumer services. Its on-line encyclopedia is updated every three months, making it better than a $1,000 set in a bookshelf that day by day grows ever more obsolete. Or we can track our investments. Or we can rap with people in the fields of religion or electronics or genealogy of pets or any other special interest. We can write a letter to anyone in the country -- or Canada -- who subscribes to GEnie. I can download TRS-80 software. It was my participation in a GEnie conference with you that convinced me to begin a subscription to TRSTimes.

Then I started contacting local bulletin board systems. Most are free. Many are IBM-oriented, but others are more general. One board I now use regularly is for freelance writers who share and critique each others' work. Another, a really super BBS, offers 260 interest categories, ranging from a host a medical topics to the TRS-80 national echo, which is a party line of TRS-80 users. Or I can send my thoughts along to talk show host Rush Limbaugh. That board costs $25 a year, but it's worth it to me because it also offers communication services to anyone in the world who has a computer that can tie into a BBS, at no additional cost.

The main purpose of this letter is to ask TRSTimes readers to look beyond the nuts and bolts of hooking up a modem and see how a modem can help ensure that the Model 4 will continue to be a useful machine for years to come.

As a matter of fact, I'd like to see more articles on how we TRSTimes readers use our TRS-80s.

Henry Blumenthal
Jacksonville, FL

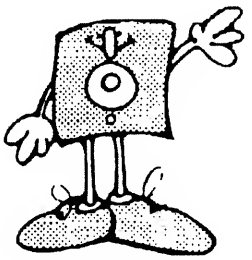
It seems we both listen to the EIB network, so to you and Rush, I say: Mega Dittoes from the Left Coast.

Your point is well taken. It would be interesting to find out how the average TRS-80 fan uses his/her machine. How about it out there!! Write to TRSTimes.

Ed.

UNDERSTANDING THE FLOPPY DISK NUMBERS GAME

By Roy T. Beck



I have long been mystified by some aspects of the numbers designating the size of some floppy drives, especially the 1.2 and 1.4 Meg varieties. Recently I took pencil to paper and worked out some tables to organize all that I knew about various drives. That may explain why the tables are so small!

As with many other aspects of this fascinating hobby of ours, today's standards are the result of both arbitrary decisions, design expedients, and the continuing advance of technology.

HISTORY OF THE FLOPPY DRIVE

The earliest floppies were the 8" variety, and the standard of the time was set by Digital Research of CP/M fame. Their 8" floppy used what is now known as Single Density encoding, using frequency modulation (FM) to put bits on the disk. Their scheme (I don't know whether they originated it or not) used two successive pulse positions along a track to form each bit position. The first pulse was always recorded. If a zero was to be recorded, then only the index pulse went on the disk, followed by a time gap where the other bit was omitted. If a one was to be recorded, then both pulses were recorded. If a continuous string of zeroes was recorded, then only the index pulses were recorded; if a continuous string of ones was recorded, then a pulse was recorded in each bit position. If the string of zero pulses are timed, they will appear at a certain rate. If the string of one pulses are timed they will appear at double the rate of the zeroes. This two to one ratio of pulses can be viewed as a modulation of frequencies, hence the encoding system was known as "FM".

ENTER THE 5.25" FLOPPY

The same encoding scheme (FM) was applied to 5.25" floppies when these came along. The original 5.25" disks had only 35 tracks. The first effort to increase the capacity of the 5.25 disks was to increase the track count from 35 to 40. In the interests of upward compatibility, the outer 35 tracks were kept in the same location and the additional tracks were added near the hub of the disk. This was fine for compatibility, as new drives could easily read either 35 tracks or 40 tracks, and the diskettes were not altered. But naturally, there had to be a fly in the ointment. Bit shift, to be exact! It seems that if you record two magnetic pulses very close together on a track, the darn things physically crawl towards each other, which then upsets the time spacing on playback. Don't ask me why this happens, I haven't the foggiest; but it does happen. To further aggravate the problem, Radio Shack in their "wisdom" omitted the known "fix" for the bit shift problem when they designed their Model I floppy disk controller system. Apparently the 35 track system was marginal to begin with, and 40 tracks was just about impossible due to the increased bit shift as the pulses in the tracks became ever more closely spaced at the higher track numbers.

Percom to the rescue! I don't know if they were first, or even if they were the only one to offer a fix for the bit shift problem, but they offered a data separator for single density users which, in my case, solved the problem. 40 tracks became reliable.

Table 1 summarizes the various TRS disk configurations and Table 2 shows the corresponding IBM information.

Table 1 - TRS Configurations

Size	Trks	Sides	Dens	Sect	Bytes	KB/sec	RPM	Capacity
90K	35	1	Sgl	10	256	250	300	89,600
100K	40	1	Sgl	10	256	250	300	102,400
180K	40	1	Dbl	18	256	250	300	184,320
360K	40	2	Dbl	36	256	250	300	368,640
720K	80	2	Dbl	36	256	250	300	737,280
720K*	80	2	Dbl	36	256	250	300	737,280

* 3.5" floppy

Table 2 - IBM Configurations

Size	Trks	Sides	Dens	Sect	Bytes	KB/sec	RPM	Capacity
160K	40	1	Dbl	8	512	250	300	163,840
180K	40	1	Dbl	9	512	250	300	184,320
360K	40	2	Dbl	18	512	250	300	368,640
800K#	40	2	Dbl	20	512	250	300	819,200
1.2M	80	2	Dbl	15	512	500	360	1,228,800
1.4M*	80	2	Dbl	36	512	500	300	1,474,560*

Homebrew configuration by NATGUG, not an IBM standard!
* 3.5" floppy

DOUBLE DENSITY

Continuing with the floppy disk story. Much later, the "double density" system came along. This system could put twice as much data in the same track space by using a different encoding scheme. I'll not try to explain it in words, pictures would be necessary. Since it was a modification of the previous single density, frequency modulation scheme, the new scheme was identified as "modified frequency modulation", or MFM. MFM actually did double the amount of data in a given distance along the track, but the requirement for additional header space along the track restricted the net gain to 80% over FM recording. Still 180% of what could previously be put on a disk was a dandy improvement.

The MFM encoding system was even more sensitive to bit shift, so the density doubler boards designed by several aftermarket groups incorporated the necessary data separator, and double density operation was solid and reliable from the beginning, and continues so today. Of course, when RS came out with a competing double density adapter for the Model I, they did it differently than the aftermarket de facto standard, and suddenly we had two double density protocols to contend with. The aftermarket DOS authors solved the problem by incorporating some code to determine which doubler (if any) was in your machine, and to then select the appropriate code to read and write with your doubler. Radio Shack was then about to bring out the Model III, and so their net contribution to the density doubling story was two more versions of the Model I DOS, those being TRSDOS 2.7DD and 2.8DD. I have never used either of them, but have been told they were really forerunners of TRSDOS for the Model III. Apparently little if any programs were offered by RS to use these DOSes, and they quickly became only historical curiousities, as the aftermarket DOSes were in control in the Model I marketplace and TRSDOS 1.0, 1.1, 1.2, and 1.3 became the RS Model III offerings.

THE TRS-80 MODEL III

The Model III TRSDOS series had a serious failing in that it could only read SS drives; even with DS drives installed, it apparently could not be patched to operate DS. This problem was tackled by another man (Gary Campbell), and his efforts produced TRSDOS 1.4 and finally SYSTEM 1.5 as aftermarket products. SYSTEM 1.5 is currently available through TRSTimes. These could handle DS drives alright, but only in a somewhat strange mode. The second sides of the drives can be written to and read, but only with separate logical drive numbers. That is, the back side of a drive gets a different number than the front. Note carefully, the disk is not a flippy. It can only be inserted in the drive one way, and in that position both sides can be read, but only as two different logical drives. If the disk is inserted wrong side to, neither side can be read. The drawback of this scheme is that the data on the backside cannot be read on a SS drive. The other person must have a similar two sided drive and SYSTEM 1.5 on his system. SYSTEM 1.5 is quite a respectable DOS, except for the method of handling DS disks and the fact that it cannot handle a hard drive.

MODEL III TRSDOS has another curious feature. The tracks are numbered from 0 to 39 as is usual, but the sectors are numbered from 1 to 18. There are no sector 0's on the disks!

The ever present desire for larger storage capacity lead to 80 track drives, with double density and double sided operation. This combination used to be known as "quad density", on the basis that it held four times the amount of data that a SS SD disk could hold; 720K vs 180K.

IBM'S EFFORTS

IBM's first floppy drive for the PC was a curiousity by present standards. It was 5.25", single sided, with eight sectors of 512 bytes each. Why eight and not nine? Gee, I dunno! A little later, IBM changed to nine sectors per track and about the same time went to double sided operation. They kept the eight and nine sector capability for compatibility, but the 360K, 40 track, DS, DD disk immediately became the standard for the PC.

HIGH DENSITY FLOPPIES

Next came some developments by IBM. Big Blue wanted more space on floppies, and they looked to see what could be done with the 5.25" floppy. They reasoned that if they could record more data per track, they could increase the total capacity of the disk. Their efforts resulted in a doubling of the data rate fed to a disk. Where all previous disks and controllers had handled data at 250 kilobits per second, IBM doubled this to 500 KB. This by itself should have allowed twice as many sectors per track, for a straightforward doubling of the data capacity of a 5.25 floppy. This would allow recording of 1.44 Megs on an 80 track, DS, DD 5.25" disk. For whatever reason, (I don't know), IBM decided this doubling of the bit rate was pushing the state of the art a little too hard. (IBM is a conservative outfit). To reduce the bit density on the disk, they increased the rotational speed of the disk somewhat, which had the effect of decreasing the bit density, measured along the track. They decided to increase the speed from 300 RPM to 360 RPM, which made the bit density 300/360 or 5/6 of the value achieved by doubling the bit rate. So, 5/6 of 2 equals 1-2/3 of the 720K capacity, for a new value of 1200K, or 1.2 Meg. The new, improved media required for this mode is the High Density floppy.

This is really only part of the explanation of what is going on in the 1.2 Meg drives. There is more there than

meets the eye, and I will delve deeper in a later paragraph. For the moment, just accept the 1.2 Meg floppy as the apparent end of the line in IBM's development of the 5.25" floppy.

NATGUG'S DEVELOPMENTS

A TRS and IBM user group in England called NATGUG (the explanation for that name is too long and dull to pursue here) has even worked out a scheme to put 800K on a 720K floppy. Their trick is to put 10 sectors of 512 bytes on the floppy, giving an 800K floppy. It requires specially written floppy drivers to function, and to me seems like a lot of effort for little profit, as the format is not supported by IBM, and interchangeability is limited to those who have the required drivers. It is probably only worth while to those lacking hard drives, which are rather expensive over there. I believe the 800K format is mostly a demonstration that it can be done, but is not widely used.

THE 3.5" SHIRT POCKET FLOPPY

Another interesting development was the introduction of the 3.5" drive. Actually, several competing companies introduced similar disks about the same time, their goals being smaller size and better protection for the disks. The present 3.5" disk is the result of the shakeout among the several competing designs. The 3.5" disk is normally an 80 track, DS, DD format, used by almost all computer manufacturers. The 3.5" drive was never used by Tandy on the Model I, III, 4 family, but the 3.5" drive can be driven by our TRS FDC's with no trouble. I have even set up a Model I booting NEWDOS 80 V.2 from a 3.5" drive, just to see if it would work. It did, with no difficulty.

THE HIGH DENSITY 3.5" FLOPPY

The pressure for more capacity is never ending. IBM (and others) next looked at the 3.5" disk. The 3.5 was really an 80 track, DS, DD disk, with identical characteristics to the 720K floppy. IBM decided to apply the 500 kilobit per second rate to the 3.5 with the best available media, and discovered it could be made to work. I can only guess the reason they could double the bit density on the 3.5 and could not quite double it on the 720K floppy must be due to improvements in the quality of the floppy media during that time period. The High Density 3.5" floppy is necessary for this operation. In any event, the 3.5" format could handle an 80 track, DS, DD format at 500 kb, which increased the capacity to 1440K, or 1.4 Meg. WOW!

Note that this development, over approximately 10 years, increased the 5.25" floppy of 35 tracks, SS, SD of 90 K to 40 tracks, DS, DD with 1200K. That is more than 13 times the original disk's capacity. An excellent example of development! And of course, the 3.5" floppy went to 1440 K, about 16 times the original.

I won't go into hard drive development, that deserves an article all its own, and I may try to write that another time.

MORE ON THE 1.2 MEG FLOPPY

Now let's see what else is going on in that 1.2 Meg drive. Remember, the drive is an 80 track drive, and has heads of the width appropriate to that standard, which is 96 tracks per inch, or 96 tpi. 40 track drives had only half or 48 tpi because their heads were wider.

Some of the TRS DOSes have provision for "double stepping" an 80 track drive so it could read 40 track disks. Since the heads in the 80 track drive are much narrower than the 40 track heads, the 80 track could easily and reliably read a 40 track disk previously written by a 40 track drive. However, writing on a 40 track disk with the narrow heads of an 80 track drive is an invitation to trouble. The narrow 80 track head cannot erase the full width of the track laid down by the 40 track head, and so you get a track on the disk analogous to a three lane highway. The "center lane" is the good data written by the 80 track head, and the two "outer lanes" contain leftover data placed by the original 40 track head. In this situation, an 80 track drive can read and write successfully.

However, if the disk is moved to another 80 track drive which is aligned slightly differently, the head of the second drive may try to read and write most of the "center lane" plus a strip of one of the "outer lanes", which can produce data errors. A still worse situation occurs if the disk written by the 80 track head is moved to a 40 track drive. Now the 40 track head is obliged to try to read the "center lane" with valid data plus the two "outer lanes" containing old data. The result will certainly be trouble.

IBM is well aware of this situation, but still felt obliged to allow their (80 track) 1.2 Meg drive to read and write, but not format, a 360K floppy. This provides upward compatibility from the original PC to the later machines. But the 1.2 Meg drive became rather schizoid in its operations. If it is to read and write on a 360K (40 track) floppy, it must be capable of the double stepping trick. Rather than do this in software, ala TRS, IBM provided another signal line between their 1.2 Meg floppy disk controller (FDC) and their 1.2 Meg drive. This line can be controlled by the FDC to tell the drive when to single step (80 tracks) and when to double step (40 tracks).

An additional fly in the ointment is the rotational speed, 300 RPM for the 360K disk vs 360 RPM for the 1.2 Meg disk. Still another control line had to be provided between the FDC and the drive.

The final discrepancy is the data rate. This was handled purely in the FDC, which can be commanded to send 250KB data or 500KB data to the drive. The drive neither knows nor cares what the data rate is.

And this is not the end of the features in the 1.2 Meg drive. Ask me some time about line 34, the ChangeLine, and the XT/AT jumper! But that is for another article another day.

1.2 MEGS IN A TRS?

Why would anyone in the TRS world care about the quirks of this 1.2 Meg IBM drive? Consider. The drive inherently has the capability of reading and writing 80 tracks at 300 RPM, which is the format of our TRS 720K disks. IBM does not support the 720K format, but the 1.2 Meg drive can do it! Some Australians have done it, and reported in various newsletters of their success. This allows them to buy readily available IBM 1.2 Meg drives and use them where scarce, practically unavailable 80 track 720K drives are needed. And we can do it, too. I notice that IBM 1.2 Meg drives are going for around $55 each, which is not a bad price, compared to 720K drives at around $90!

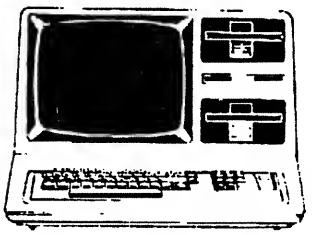
The Australians worked out jumper settings for several different drives which in essence locked the two lines controlling RPM and single/double stepping into the 300 RPM and single stepping mode. Having accomplished this, the drives could then be driven at 250 KB by our normal TRS FDC's, and 80 track drives are now readily available from the IBM parts bins. How to install and correctly jumper a 1.2 Meg IBM drive in a TRS will be easy, once the jumpering is sorted out and published. Since I have only second hand reports on only two Panasonic drives to work from, I will defer this task to a future article, when I get "a round tuit".

Meantime, file this article in your collection, it may be useful in dealing with "strange" (IBM) drives.

Checkbook

Model III - Basic

By J. R. Nock



This is a BASIC program for the Model III that I have used for years to help keep my check book balanced. Entries can be made as often as you like, to check the arithmetic. At the month end the bank's balance and the cancelled checks are entered and compared to your input.

The outstanding checks and the balance are stored to disk, and optionally can be printed out. The input outstanding checks are saved to a separate file for emergency recovery if the file is wiped out by accident.

A special subroutine is used for keyboard input. It checks for numbers or characters, and indicates the space available in each input field.

```
1 REM- PROGRAM CHECK - CHECK BOOK
BALANCER  STORES OUTSTANDING CHECKS ON
DISK :0 -  J. R. Nock, 1992
5  CLEAR 8000
10 DEFINT I-N
19 K=150
20 DIM A$(K), NO(K), IDATE(K), AMT(K)
30 TI$="check #    date       payee              amount
balance"
31 TJ$="check #    date       payee              amount"
95 C$="$$##,###.##-"
100 CLS:PRINT @ 15,"*** CHECK BOOK BALANCER -
MAYBE ***":PRINT
205 REM: RETURN IF INPUT INCORRECT
210 CLS:PRINT @ 0," IF THIS IS THE FIRST USE OF
THE PROGRAM, ENTER YOUR BALANCE AT THE
START OF THE PERIOD": PRINT
220 PRINT " OTHERWISE JUST PRESS 'ENTER'":
230 FL=-9:GOSUB 32100: BAL#=VAL(IN$)
240 PRINT: PRINT "NOW ENTER THE FINAL BANK
BALANCE FOR THE PERIOD ";: FL=-9:GOSUB 32100:
BBAL#=VAL(IN$):PRINT
250 PRINT:PRINT "CHECK THE BALANCES ENTERED
CAREFULLY."
251 PRINT:PRINT"ARE THEY ALL CORRECT (Y/N) ? ";:
FL=1:GOSUB 32100:IF IN$ < >"Y" THEN 205
260 B#=BBAL#
300 IF BAL#= 0  GOTO 1000
320 GOTO 1500: REM ** THIS IS FOR THE FIRST
TIME RUN ONLY
1000 REM ** NORMAL RUN - READ THE PREVIOUS
MONTHS CHECKS **
1005 PRINT:PRINT "DISK SHOULD BE RUNNING"
1010 I=0: OPEN "I",1,"CHEX/TXT:0"
1020 INPUT #1, IDATE(I), NO(I), AMT(I), A$(I):
IF IDATE(I)=0 THEN 1040
1030 IF EOF(1) THEN 1040 ELSE
PRINT NO(I), IDATE(I), A$(I), AMT(I):
I=I +1: GOTO 1020
1040 BAL#= AMT(I): CLOSE 1:
PRINT "BALANCE FROM DISK= ";USING C$;BAL#
1050 PRINT "PRESS ENTER TO CONTINUE ";:
FL=1: GOSUB 32100
1500 REM **  READ FROM DISK DONE - GET CHECK
BOOK DATA **
1550 CLS
1551 PRINT @ 0, " ":
PRINT TAB(20) "*** CHECK BOOK ENTRIES ***":
PRINT "FOR DEPOSITS MAKE CHECK # = 0"
1554 PRINT "ENTER CHECK # '-1' TO DELETE THE
LAST ENTRY, '9999' TO FINISH"
1560 PRINT@256,TI$: 'PRINT A COLUMN TITLE
1600 PRINT@576,TJ$
1602 PRINT@640,"";:FL=-4:GOSUB 32100:
E=VAL(IN$):IF E<0 THEN 1605 ELSE
IF E=9999 THEN 1608 ELSE NO(I)=E:
GOTO 1606
1604 IF E<0 THEN 1605 ELSE IF E<101 THEN 1602
ELSE IDATE(I)=E:
GOTO 1606
1605 I=I-1 : BAL#=BAL# - AMT(I):
PRINT@448,CHR$(255): GOTO 1602
1606 PRINT@652,"";:FL=-4:GOSUB 32100:
E=VAL(IN$):IF E< 0 THEN 1606
ELSE IF E<101 THEN 1606
ELSE IDATE(I)= E: GOTO 1610
1608 PRINT@768,"DID YOU ENTER BANK CHARGE,
ANY AUTOMATIC WITHDRAWALS, INTEREST?":
PRINT @832 ,"  (Y/N) ? ";:
FL=1:GOSUB 32100:
PRINT @768,CHR$(255):
PRINT@832,CHR$(255)
1609 IF IN$="Y" THEN 1700 ELSE 1602
1610 PRINT@662,"";:FL=10:GOSUB 32100:A$(I)=IN$
1611 PRINT@684,"";:FL=-9:GOSUB 32100:
AMT(I)=VAL(IN$):IF AMT(I)< .005 THEN 1611
1617 IF I<1 GOTO 1645
1620 PRINT@384,CHR$(255):PRINT @ 384,NO(I-1):
PRINT @ 395,IDATE(I-1)
1630 PRINT @ 405, A$(I-1): PRINT @ 423,"";:
PRINT USING C$;AMT(I-1)
1645 IF NO(I) > 9998 GOTO 1650:
REM* NO PRINT ON FIRST INPUT
1646 IF NO(I) > 0 THEN AMT(I) = -AMT(I)
1647 BAL#= BAL# + AMT(I)
1650 PRINT@448,CHR$(255):
PRINT @ 448,NO(I):
PRINT @ 459,IDATE(I)
```

```
1660 PRINT @ 469, A$(I): PRINT @ 487,"";
1661 PRINT USING C$; AMT(I)
1670 PRINT @ 500,"";
1671 PRINT USING C$; BAL#
1675 IF NO(I) > 9998 THEN 1700
1676 IF I<148 THEN 1685 ELSE
PRINT"*** ARRAY IS FULL ***":
GOTO 1700
1685 I=I+1:PRINT @ 576,"                              ":
FOR X= 1 TO 4: PRINT "                                ":
NEXT X:GOTO1600
1700 REM *** FINISHED WITH THE STUBS - NOW
READ THE CANCELLED CHECKS
1800 CLS
1810 PRINT @ 20,"*** CANCELLED CHECKS ***":
PRINT "FOR DEPOSITS MAKE CHECK # = 0"
1820 PRINT"ENTER DATE 9999 TO SHOW ALL
CHECKS ARE ENTERED"
1830 PRINT@256,TJ$
1850 PRINT@576,TJ$
1855 PRINT @640,CHR$(255):PRINT@640,"";:
FL=-4:GOSUB 32100:
K=VAL(IN$):
IF K>9998 THEN 1870
1856 PRINT@650,"";:FL=-4:GOSUB 32100:
J=VAL(IN$):IF J<0 THEN 1856
1858 PRINT@680,"";:FL=-9:GOSUB 32100:
B=VAL(IN$):IF B<1 THEN 1855 ELSE 1900
1870 PRINT@768,"DID YOU REMEMBERTO ENTER
THE DEPOSITS, TRANSFERS, INTERESTINTERBANK
(IB) TRANSACTIONS (Y/N) ?";:FL=1:
GOSUB 32100:
PRINT@768,CHR$(255):
PRINT@832,CHR$(255)
1871 IF IN$="Y" THEN 2100 ELSE 1855
1890 REM *** COMPARE ENTRY WITH SAVED
CHECK STUBS
1900 FOR L=0 TO I-1
1910 IF K<> NO(L) GOTO 2050
1920 IF J <> IDATE(L) GOTO 2050
1930 IF K>0 AND (ABS(AMT(L)+B))>0.003
THEN 2050
1940 IF K=0 AND (ABS(AMT(L)-B))>0.003 THEN 2050
1950 PRINT@384,CHR$(255):PRINT @ 384, IDATE(L)
1951 PRINT@ 395,NO(L)
1960 PRINT @ 405, A$(L): PRINT @ 423,"";:
PRINT USING C$; AMT(L)
1970 PRINT @448,STRING$(63," "):
PRINT@458,"STUB AGREES WITH CANCELLED
CHECK"
1980 IDATE(L) = -IDATE(L)
1990 GOTO 1855
2050 NEXT L:
PRINT@384,"*** THE STUB DATA DOES NOT AGREE -
PRESS ENTER TO TRY AGAIN **":
PRINT " OR ENTER '9' TO GO BACK TO THE CHECK
STUBS ";:FL=-1:GOSUB 32100:J=VAL(IN$):
IF J=9 THEN 1550
2090 GOTO 1800
2100 REM ** PRINT OUT OUTSTANDING CHECKS **
2110 CLS:
PRINT @ 15," OUTSTANDING DATA "
2121 PRINT @ 128," DATE ":
PRINT @ 137,"CHECK #":
PRINT @152,"PAY TO":
PRINT@172,"AMOUNT":
PRINT@185,"KEY #"
2122 JN=1:
FOR L=0 TO I-1:
IF IDATE(L) <0 GOTO 2150
2125 BBAL#= BBAL# + AMT(L)
2130 PRINT IDATE(L);TAB(11) NO(L); TAB(21) A$(L);
TAB(39) "";
2139 PRINT USING C$; AMT(L);
2140 PRINT TAB(58) L
2145 JN=JN+1:
IF JN <>10 GOTO 2150
2146 PRINT "TO CONTINUE THE LISTING PRESS
'ENTER'";:
FL=0:GOSUB 32100:
JN=1:PRINT
2150 NEXT L
2160 PRINT "                - YOUR BALANCE= ";
2161 PRINT USING C$; BAL#
2163 PRINT "                - BANK BALANCE= ";
2164 PRINT USING C$; BBAL#
2166 PRINT "                -  DIFFERENCE= ";
2167 PRINT USING C$; BBAL# - BAL#:
PRINT " PRESS 'ENTER' TO CONTINUE ";:
FL=0:GOSUB 32100
2169 CLS:PRINT " OPTIONS:" : PRINT
2170 PRINT "< 0 > RECORD THE OUTSTANDING
CHECKS TO DISK< 1 > RETURN TO CORRECT THE
CHECK BOOK STUBS< 2 > RETURN TO CORRECT
THE CANCELLED CHECKS OR DEPOSITS< 3 > RE-
PEAT THE
2171 PRINT"<4> LIST THE OUTSTANDING CHECKS
ON THE PRINTER ( READY IT":PRINT"< 5 > DELETE
A BAD CHECK STUB ENTRY< 6 > END THE JOB (BE
SURE YOU HAVE SAVED TO DISK!)":PRINT:
PRINT "YOUR CHOICE ? ";
2172 FL=-1: GOSUB 32100: E=VAL(IN$):
IF E>6 THEN 2172
2173 ON E+1 GOTO 2200, 2174, 2174, 2174, 2700,
     2180, 2500
2174 BBAL#=B#: ON E+1 GOTO 2200, 1550, 1800,
     2110, 2700, 2180, 2500, 2500
2176 REM DELETE BAD ENTRY IN CHECK BOOK
2180 CLS:PRINT" WHAT IS THE BAD ENTRY KEY
NUMBER? ";:FL=-3:
GOSUB 32100: E=VAL(IN$)
2181 PRINT:PRINT:PRINT NO(E);
2182 PRINT TAB(11) IDATE(E);
2183 PRINT TAB(21) A$(E);
2184 PRINT TAB(39) "";
2185 PRINT USING C$; AMT(E);
```

```
2186 PRINT TAB(58) E: PRINT "IS THIS THE ENTRY TO
DELETE?  (Y/N) ";
2187 FL=1:GOSUB 32100: IF IN$="Y" THEN 2196
ELSE 2169
2195 IF X < > 9 GOTO 2169
2196 IDATE(E) = - IDATE(E): BAL# = BAL# - AMT(E):
GOTO 2169
2200 CLS: REM * STORE OUTSTANDING CHECKS ON
DISK 1
2210 OPEN "O",1,"CHEX/TXT:0"
2335 JN=0
2339 I9= I -1:Y$="*"
2340  FOR L=0 TO I9: IF IDATE(L)< 0 GOTO 2400
2350 PRINT"OUTPUT TO DISK "; IDATE(L); NO(L);
A$(L); AMT(L)
2360 PRINT #1, IDATE(L); NO(L); AMT(L); A$(L)
2400 NEXT L
2410 V=BBAL#: PRINT #1, 0; 0; V; "BALANCE"
2420 CLOSE 1
2430 GOTO 2169
2500 CLS:PRINT:
PRINT "THAT SHOULD BE ALL FOR THIS MONTH"
2510 PRINT "WASN'T THAT FUN !!!"
2520 PRINT "SEE YOU IN 30 DAYS"
2600 END
2700 REM * LIST ON PRINTER
2705 IF PEEK(14312) = 63 THEN 2710 ELSE PRINT:
PRINT"** PRINTER IS NOT READY - ENTER 'Y' TO
TRY AGAIN, 'N' TO SKIP":PRINT" ENTER (Y/N) ":
FL=1: GOSUB 32100:
IF IN$="N" THEN 2169 ELSE 2705
2710 LPRINT "          OUTSTANDING CHECKS  ";
LEFT$(TIME$,8): FOR L=0 TO I-1:
IF IDATE(L)<0 THEN 2750
2720 LPRINT IDATE(L);TAB(11) NO(L);TAB(21) A$(L);
TAB(39) "";:LPRINT USING C$; AMT(L)
2750 NEXT L: GOTO 2169
3000 INPUT "AT 3000, INPUT "; X$
3005 GOSUB 5000
3010 PRINT X$,Y$
3020 GOTO 3000
5000 REM *** PAD OUT STRINGS TO 10,
CONCATENATE INTO Y$
5010 LL=LEN(X$): IF LL< 10 GOTO 5050
5020 IF LL= 10 GOTO 5100
5030 PRINT "*** ERROR - MUST TRUNCATE "; X$
5040 X$ = LEFT$(X$,10)
5041 GOTO 5110
5050 FOR I=1 TO 10-LL:
X$= " "+X$:
NEXT I
5100 Y$= Y$ +X$
5110 RETURN
5120 LL= LEN(X$):
IF LL<10 GOTO 5150
5125 IF LL=10 GOTO 5100
5130 GOTO 5040
5150 FOR I=1 TO 10-LL:
X$=X$ + " ":
NEXT I
5160 GOTO 5100
32000 REM KEYBOARD INPUT SUBROUTINE
32100 IN$="":W$=INKEY$:W=14:
WD=0:WS=WD:WL%=WD:
IF FL=WD THEN FL=1
32105 PRINT STRING$(ABS(FL),136);
STRING$(ABS(FL),24);
32110 PRINT CHR$(W);:
FOR W%=1 TO 25:
W$=INKEY$:
IF W$< >"" THEN 32115 ELSE NEXT:
PRINT CHR$(15);:
FOR W%= 1 TO 25:
W$=INKEY$:
IF W$< >"" THEN 32115
ELSE NEXT:
GOTO 32110
32115 PRINT CHR$(W);:
IF ABS(FL)=WL% THEN 32125 ELSE
IF FL>0 AND W$> =" " AND W$< ="Z" THEN 32170
ELSE IF FL <0 AND W$>"/" AND W$<":" THEN 32170
32116 IF W$> =CHR$(97) AND W$< =CHR$(122)
THEN W$=CHR$(ASC(W$)-32):
GOTO 32170
32117 IF W$="," THEN PRINT W$;:
WL%=WL% +1:
GOTO 32175
32120 IF W$="." AND WD=0 THEN WD=1:
GOTO 32170
32123 IF(W$="-" OR W$=" +") AND WS=0 AND
WL%=0 THEN WS=1:
GOTO 32170
32125 IF W$< >CHR$(8) THEN 32150
ELSE IF WL%=0 THEN 32110 ELSE PRINT CHR$(24);:
IF FL>0 THEN 32135 ELSE
IF PEEK(16418)=44 THEN 32140
32130 IF PEEK(16418)=46 THEN WD=0:
GOTO 32135 ELSE IF PEEK(16418)=43 OR
PEEK(16418)=45 THEN WS=0
32135 IN$=LEFT$(IN$,LEN(IN$)-1)
32140 WL%=WL%-1:
POKE 16418,136:
GOTO 32110
32150 IF W$=CHR$(24) THEN
PRINT STRING$(WL%,CHR$(24));:
GOTO 32100
32155 IF W$< >CHR$(13) THEN 32110 ELSE
PRINT STRING$(ABS(FL)-WL%,32);
32160 PRINT CHR$(15);:W%=25:
NEXT: RETURN
32170 PRINT W$;:
IN$=IN$+W$:WL%=WL%+1
32175 IF ABS(FL)=1 THEN 32160 ELSE 32110
32180 END
```

Recreational and Educational Computing
Finding the Next Number

(c) 1992, Dr. Michael W. Ecker, TRSTimes Contributing Editor

Very popular in recreational math are those "What's the next number" questions. You know the type. You are given something like: 1, 2, 4, ? You are supposed to find some pattern that links and generates the first few numbers, and then you extrapolate to get the next number based on said pattern.

Sometimes the questions are not so obvious. Suppose you consider the number sequence: 2, 6, 12, 20, ? What is the next number now? If that is still too easy, how about: 10, 5, 1, -3, ? What is the next number now?

Let's start with the first question, if only because it is so ostensibly easy.

If you said the answer is 8, then you saw the pattern of doubling. You must be saying: "Mike, come on now. This is too easy! Are you teasing us?"

However, is it not also possible to see another pattern? For instance, might the pattern not be to first add 1 to the first number (1) to get 2, add 2 to the second number (2) to get 4, and therefore now to add 3 to the third number (4) to get 7? Is this not just as good an answer?

For a solution, then, one need only find any pattern (alternatively: rule, sequence, or function) that produces the first few values. Then, whatever said rule produces for the next number is a correct answer. In fact, in 1953, mathematics professor Jim Householder published a piece entitled "Note to a Psychologist" in which he did something startling. He gave an explicit means of constructing a formula, in advance, that not only produces any prescribed or given first N values, but also yields any arbitrarily desired result for the (N+1)st value. As luck would have it, Jim became a subscriber to my own **Recreational & Educational Computing (REC)** publication some years ago. So, when I broached this topic in my final "Recreational Computing" column in **Creative Computing** magazine (which magazine folded Dec. 1985) and later in my own publication (early 1986), Jim sent a copy of this article, and he kindly allowed me to reproduce it in **REC** later (late 1980s).

The question "Find the next number" is a fraud, a hoax, a meaningless exercise. It is perpetuated by well-meaning math teachers who intend to find interesting puzzles. Unfortunately, they are evidently unaware that the exercises posed are meaningless utterances of questions not really well-defined.

Here is the paradox. As we saw in the first, easy example, there are multiple patterns possible. So, in order to be fair, posers of such "next number" questions would have to spell out what they mean. That, in turn, would defeat the purpose of the question, which we may presume to be some measurement of ability to discern a pattern.

In conclusion, we really should speak of "a next number" and not "the next number" (notwithstanding the fashionability of the latter).

Not all is lost; we can salvage something from all this. For, despite the disappointment of learning that the question itself is virtually meaningless - appearances notwithstanding - we can make sense out of this if we are willing to define the rules of the game. This does render this subject more mechanical, but at least you'll know what all the fuss was about on IQ and other tests.

Let's take the example of: 2, 6, 12, 20, ? One standard technique is to see whether there is a constant difference between consecutive terms. The respective differences (latter minus former) are: 4, 6, 8 (from 6-2 = 4, 12-6 = 6, and 20-12 = 8). Hmm... No good, but note that these differences themselves have a common difference of 2! So, we can take their differences and picture this as:

```
2 6 12 20
 4 6 8
  2 2
```

Now we can extend this picture by working in reverse. Instead of working from the top down, we now addend another 2 to the bottom row and work from the bottom up. Instead of subtracting, of course, we must now add as we go up. This produces this new diagram obtained from the bottom right and working up:

```
2 6 12 20 30
 4 6 8 10
  2 2 2
```

Of course, some of you may have actually recognized that I may have been thinking of the formula f(n) =

n(n+1). In other words, each number is obtained by multiplying the term number by one more than said term: 1x2, 2x3, 3x4, 4x5, etc., so it is no surprise to get 5x6 as next number. Incidentally, this sequence represents the so-called oblong numbers, numbers that represent the sum of the first n even numbers. (E.g, sum of first 5 evens is 5x6 = 30.)

Let's take the crazy-looking example of: 10, 5, 1, -3, ? now:

```
10 5 1 -3
 -5 -4 -4
   1 0
    -1
```

Now we again extend this picture by working in reverse:

```
10 5 1 -3 -8
 -5 -4 -4 -5
   1 0 -1
    -1 -1
```

If you check, you will see that the differences of terms in any row do indeed generate the row beneath it, although you do have to know how to subtract signed numbers. If you're desperate, you can stoop to using a calculator or a computer, but I hope that you don't find that necessary!

We can write a program to generate a next number under the rule that an appropriate row of differences is constant. In practice, this boils down to two possibilities: Either some row does have the same entry (as in the earlier triangle diagram) or we get down to some solitary number (as in last example).

The program should allow you to specify the number of original terms, N, and input the first N numbers. To get term number (N+1), I effectively mimic the triangle approach, namely by using a doubly-subscripted array A(J,K) for the entry in the j-th row that is number k in that row. I will omit programming details unless readers write to request more information on this. Just remember that the next number this BASIC program produces is consistent with the formulation of an eventual constant difference. Thus, it will give 7, not 8, as its answer for our original question of next number for: 1, 2, 4, ?

```
10 CLS: Print "Dr. M. Ecker's 'Finding A Next Number',
(c) 1985, 1987, 1992."
20 PRINT "This program will allow you to input virtually
any number of numbers"
30 PRINT "and then determine the next number. This
number, of course, is based"
40 PRINT "on an eventual N-fold difference of zero."
50 PRINT
60 INPUT "<ENTER> to begin..."; XX$: PRINT: PRINT
70 INPUT "How many numbers are given?"; N
80 DIM A(N+1,N+1)
90 FOR J=1 TO N
100 PRINT "Element number "; J;
110 INPUT A(1,J)
120 NEXT
130 FOR K=2 TO N
140 FOR J=1 TO N+1-K
150 A(K,J)=A(K-1,J+1)-A(K-1,J)
160 NEXT J
170 NEXT K
180 A(N,2)=A(N,1)
190 FOR L=N-1 TO 1 STEP -1
200 A(L,N-L+2)=A(L,N-L+1)+A(L+1,N-L+1)
210 NEXT L
220 PRINT "The next number is ";
230 PRINT A(1,N+1)
```

As a FREE service I offer this program on disk with no obligation to any reader who sends me: 1) a pre-formatted diskette (TRS-80 Model 3 TRSDOS; or blank cassette tape for TRS-80 Model 100 or 102 laptop; or any PC/ MSDOS format at all, any disk size, either in BASIC or compiled to run without GWBASIC; or Sanyo 555; or Macintosh 800K or 400K); and 2) a self-addressed return mailer bearing your full name and address *and* 75 cents' postage affixed. In the alternative, send me just $3 alone and I'll send this program and four more on disk (slot machine, fraction addition, organ, collection completion).

For those interested in a fuller collection of a dozen more graphical recreations and programs on disk for TRS-80 Models 3 and 4, I have a Logic Games Disk available for $10 plus $2 shipping. Or take above disk and this one for $13.50. Send orders to Recreational Mathemagical Software/ 909 Violet Terrace/ Clarks Summit, PA 18411.

Special Offer

I'll spare you the high-pressure sell and come to the point. Try one year of *Recreational & Educational Computing* ($27) and you may have your choice of:

- FREE, the above Logic Games Disk plus Next Number program for Model 3 or 4.
- FREE, professionally written commercial blackjack program, Casino 21 for the TRS-80 (Model 1, 3, or 4, tape or disk) - ordinarily $16.95, or:
- Magic Math Plus, available with special versions for TRS-80, PCs, Sanyo 555, Apple II, and Macintosh. It is comprised of nearly 40 "mathemagical" programs in the TRS-80 double-volume, all menu-driven on five menus on a self-booting disk for just $9.95 plus $1 shipping (instead of $40).

If **REC** does not meet your expectations, you may cancel your subscription and receive a no-questions asked prorated refund on all unmailed issues, yet still keep your free or heavily-discounted software.

To take advantage of this offer, you MUST mention this publication and mail your prepaid order by Sept. 30, 1992 (or two months after this issue is published, whichever comes later). Please send me your comments, questions, solutions, improvements, programs (on disk as well as paper), and your orders for **Recreational & Educational Computing**. Write to:

Dr. Michael W. Ecker
TRSTimes' "Recreational Computing"
909 Violet Terrace
Clarks Summit, PA 18411

Until next time, Happy Recreational TRS-80 computing!
Mike

****
****

Dr. Michael W. Ecker is a Penn State University math professor as well as a computer writer-reviewer and columnist with 350 publication credits. Mike is also Editor/Publisher of **Recreational & Educational Computing (REC)** and the TRS-80 columnist for Vulcan's **Computer Monthly**, the only major computer magazine with a TRS-80 column.

REC, from which these articles have been adapted, is in the middle of its seventh year and is available for $27 per calendar-year of 8 issues, prepaid. It focuses on "mathemagic" and computer recreations. Readers are also invited to try a trial subscription of three issues for $10.

HINTS & TIPS

MultiDOS 2.x Error Codes
By Jim King

BASIC

0	NF	NEXT without FOR
2	SN	Syntax Error
4	RG	RETURN without GOSUB
6	OD	Out Of Data on READ or INPUT#
8	FC	Illegal Function Call: N(-#),Sqr(-#),Log(0),--
10	OV	Overflow or Underflow
12	OM	Out of Memory
14	UL	Undefined Line
16	BS	Subscript Out of Range
18	DD	Redimensioned Array
20	/0	Divide by 0
22		Undefined USR function
24	TM	Type Mismatch, e.g. Integer to $tring
26	OS	Out of $tring space
28	LS	$tring longer than 255 bytes
30	ST	$tring too complex
32	CN	Cannot Continue
34	NR	No RESUME
36	RW	RESUME without ERROR
38		Unknown
40	MO	Missing Operand
42		Exit without FOR
44		User abort
46-98		Unknown

DOS

100	Field Overflow 255 bytes for random
102	Disk I/O, CMD"E
104	Buffer # not available, or used improperly
106	File Not Found
108	Bad File Mode, e.g. Seq on Random
110	File already open
112	Disk Read, CMD"E
114	Disk Write, CMD"E
116	Wrong Password/File already exists
118	EOF encountered
120	Drive Not available
122	Disk Full
124	EOF reached before anything read
126	Attempted to access record 0/Bad PUT parameter
128	Bad Filename
130	Access mode differs from OPEN mode
132	I/O buffer overflow
134	Directory Full
136	Disk Write Protected
138	Password Protected
140	Directory full; File cannot be extended
142	No buffer; File not OPENed
144	Undefined
146	File Not Found
148	Unknown Error

THE JITTERS
By Frank Gottschalk

My Model 4 suddenly developed the "jitters" last week. A couple months ago, the screen was shrinking and blinking. At that time, suspecting a poor power lead connection, I found it by wiggling wires to the video board and found the problem in the connector at the power supply.

I pulled the contacts out of the connector and cleaned them with an eraser. Then I cleaned the contacts on the power supply pins of all three connectors. Finally I put the video connector in a different place. Success! The video was rock sollid again.

Last week, many strange things happened. I've always had an intermittent boot problem with that machine, but now, when it did boot, two drive lights would flash and sometimes settle on the right one and boot. Then suddenly strange characters would appear out of nowhere and fill up the command line! Touching some keys would produce a string of that character plus some garbage symbols. This looked serious!

I started by putting my "scope" on the power leads to the mother board and got lucky right away. The 5V line was only 4.5V and was jittering between 4.2V and 4.7V. Suddenly the strange characters started appearing in time with the jittering.

Having read about poor solder joints in the M4 power supplies, I pulled it and resoldered the connector joints, but no improvement. More probing with the scope showed me it was faulty between the connector contact pin and the wire to the motherboard connector.

I proceded to pull that pin out of the connector and soldered the crimped connection. Success! Everything is rock solid again, and it hasn't failed to boot everytime since, for the last week and a half.

As is the case so often, problems stem from faulty, aging connections.

MENU FOR NEWDOS/80
Model I/III
By Lance Wolstrup

For the Model I/III NEWDOS/80 devotees, here is a little goodie to make life somewhat little easier. The program, ND80MENU/BAS, reads the directory of drive :0, creates a menu from the directory entries, and then allows you to select a /CMD or /BAS file to execute.

This is a 'shortie', but don't let that fool you. It is a good example that a program does not have to be long and complicated to do useful things.

Type in ND80MENU/BAS and enjoy the easy way of running programs from NEWDOS/80.

```
10 'ND80MENU/BAS
20 'for NEWDOS/80 - Model I and III
30 '
100 CLEAR 500:DIM A$(26)
110 CMD"DIR"
120 SC=15488
130 FI=0:PRINT@64,CHR$(30)"CREATING MENU";
140 IF PEEK(SC)=32 THEN 200
150 FOR X=0 TO 12:
IF PEEK(SC+X)=32 THEN X=12:GOTO 160
ELSE A$(FI)=A$(FI)+CHR$(PEEK(SC+X))
160 NEXT
170 FI=FI+1:IF FI/4=INT(FI/4) THEN SC=SC+19
ELSE SC=SC+15
180 IF FI>26 THEN 200 ELSE 140
200 PRINT@0,CHR$(31):
PRINT@25,"NEWDOS/80 MENU"
210 Y=130
220 FOR X=0 TO FI-1
230 PRINT@Y,CHR$(64+X)") "A$(X)
240 IF X/3=INT(X/3) THEN Y=Y+24 ELSE Y=Y+20
250 NEXT
260 Y=INT(Y/64)*64+130
270 PRINT@Y,"YOUR SELECTION (@-"CHR$(63+FI)")"
280 I$=INKEY$:
IFI$<"@" OR I$>CHR$(63+FI) THEN 280
ELSE I=ASC(I$)-64
285 PRINT@Y,CHR$(30);"ATTEMPTING TO EXECUTE "
A$(I)
290 IF RIGHT$(A$(I),3)="CMD" THEN CMD A$(I):
GOTO 200
300 IF RIGHT$(A$(I),3)="BAS" THEN RUN A$(I):
GOTO 200
310 PRINT@Y,CHR$(30)"UNABLE TO EXECUTE "
A$(I)" - PRESS <ENTER>";
320 I$=INKEY$:
IF I$<>CHR$(13) THEN 320 ELSE 200
```

THE CR-LF PROBLEM
By Roy T. Beck

I have a single printer connected to both an IBM clone and a Model 4P via an A-B Switch. The switch works fine, thank you, but the printer has a touch of schizphrenia. It can't be sure whether it is supposed to generate a line feed (LF) after every incoming carriage return (CR) or whether it is supposed to play stupid and let the computer send LF's when necessary. And that printer is not very bright, fellows! The obvious answer is to shift the internal DIP switch whenever I change computers, thus sparing the printer the task of deciding for itself.

But my printer is old, and the DIP switches are buried inside it. If I use a flashlight, I can just see the DIP switch through a crack in the case, and it is impractical to shift the switch casually when I change computers. What to do? I solved this problem a long time ago, (the printer is about 7 years old), but I thought some of you might be interested in knowing how I did it.

I examined the DIP switch carefully, and discovered it is simply an SPST switch, the simplest type made. But I certainly could not physically relocate it, as it is securely attached to the PC board and is part of a group of 8 switches. Viewing it simply as a switch, if I left it always open, I could parallel it with a toggle switch located in a more accessible location. Accordingly, I bought a small bat handle toggle switch and mounted it in a hole in the plastic case. I then soldered two wires between corresponding terminals on the old and new switches. Now the external switch handle allows me to select between the two modes, and when I shift the A-B switch to change computers, I also move the bat handle toggle, and presto, everything comes out right.

The idea is simple, costing only the price of a switch and a foot of wire, and my computers and printer have worked properly ever since I bought the clone.

THE PRODUCER REFERENCE GUIDE
from the TRSTimes vault

The meterial featured below has been placed in the public domain and is presented here to help newcomers to the TRS-80 world getting started with the PRODUCER, a fine program that will write a data base according to the user's specifications.

To select a command from the Main Menu, press the number in front of the command. Do not press <ENTER>.

(1) ADD a new record to the file

(a) Use all Screen-Oriented Data Entry techniques.
(b) Press <=> at beginning of Field to duplicate the same Field from the previous record ADDed or EDITed.
(c) Press <@> in any Field to Save Record; Press <ENTER> in the last Field to Save Record.
(d) Press <CLEAR> in any Field to see PROMPT for the Field that you are in at the time.

(2) EDIT/Display a Record in File
(Also REPLACE and NEW)

(a) Enter any part or all of the KEY Field from left to right. E.g.: If any KEY is ROGER, you can enter <R> or <RO> or <ROG> or <ROGE> or <ROGER>. The more you enter, the better chance of finding the record you want.
(b) Enter Secondary KEY in the same manner. If you enter a secondary KEY, then only a record match-

ing the main KEY and the Secondary KEY will be displayed.

(c) If you enter no Secondary KEY then you will be asked for each matching Main KEY if "CORRECT ITEM (Y/N)?". Answer with a <Y> for Yes, or <N> for no.

(d) When record is displayed, your cursor will appear in the first Field of the record.

(e) You may EDIT any INPUT Fields using the Data Entry.

(f) To DELETE a record, type DELETE in the first six positions of the KEY field and press <@>. See option (b) on the Options Menu Selection.

(g) If you change the KEY field, you will be prompted on exit as follows:

(R)EPLACE, (N)EW, (E)ND?

(R) will write displayed Record on top of existing Record--BTREE KEYS are UPDATED. Replacement is used to Update or Edit a KEY Field.

(N) will create a NEW Record. This option is used to create Multiple Keys for the same record.

(E) will END and go back to the Main Menu.

(3) PRINT a variety of REPORTS from Report Menu.

(a) Up to 9 Report options will be displayed.

(b) Press number beside desired Report.

(c) Follow the prompts carefully.

(d) To Abort Report in progress, press <@> several times.

(e) When SORT prompt comes up, answer <N> if file is same as last time report was run, or <Y> if changed.

(4) OPTIONS Menu (Other Menu Choices)

Choose any of these options from this menu by pressing the number shown or the letter in parentheses. You may also select option from the Main Menu by pressing Letter.

[1] (D)elete a Record. Follow carefully the Prompts in the Error Message Area.

[2] (S)how a range of Records based on your SEARCH criteria.
Enter Field # to Search on when prompted.
Choose 1 of 7 options below for Search Criteria.

<1> EQUAL TO ##
<2> GREATER THAN #### SEARCH FOR STRING FROM LEFT
<3> LESS THAN ##
<4> SEARCH FROM ANY PLACE IN FIELD (STRING)
<5> EQUAL TO ##
<6> GREATER THAN #### SEARCH FOR NUMERIC VALUE
<7> LESS THAN ##
Enter Search information in field when prompted.
Select (1) for start of file or (2) for end of file.

[3] (R)eplace contents of Field Globally and Recalculate
Enter Field # whose contents you want to Replace.
Type Replacement Information in Field when prompted.
Answer <Y> to do, or <N> not to do Calculations..

If Restriction is desired, follow prompts, and refer to the (S)how Option for details.

(5) EXIT from the Program to BASIC.

The DATA ENTRY Commands

RIGHT ARROW	Move cursor one space right
LEFT ARROW	Move cursor one space left
SHIFT RIGHT ARROW	Insert space at cursor position
SHIFT LEFT ARROW	Delete a character at cursor position
SHIFT UP ARROW	Move cursor to start or end of line
@	Exit and save all data
ENTER	Move to next field below or to right. If in last Field, Exit and save data.
CLEAR	Display custom prompt for Field
SHIFT DOWN ARROW	This is defined as the CONTROL KEY.
CONTROL G	Goto a selected Field. You will be Prompted for Field in Error Message Area.
=	Duplicate in current Field the same Field from last accessed Record.

ERROR MESSAGES in DATA ENTRY

BOUNDARY LIMIT	Attempt to go outside of range of allowed Fields or outside the limit for maximum number of characters in a Field.
ILLEGAL CHARACTER	Tried to enter illegal character in Field. Cursor stays in same place.
RESULT TOO LARGE	Result of Calculation does not fit the display format that was defined
DIVISION BY ZERO	Attempt to divide by zero in a calculation (usually caused by leaving a blank Field for one that was to be used in calculations..
UNKNOWN ERROR	Just that, an Unknown Error.

CHECKING OUT DR. PATCH

Review by Henry H. Herrdegen

I was still looking at the announcement in the last issue of TRSTimes (5.2), and contemplating if this really could be worthwhile. An old patcher like me, and having given up on writing something similar to my 1.3 PUP disk for the 6.3.0, because of the difficulties with JCLs, I was plain curious.

Then, surprise, a complimentary disk arrived, with a request for me to review it. Sure! One does not get the chance very often to put a new TRS-80 program through the wringer.

Enough preamble, short and sweet: it is great. If you have read the announcement and the lead-in the Little Orphan Eighty column, then you know that it has 32 patches and their reversals, and basically what it does. All 64 work as promised (what else did you expect?). It is all written in plain BASIC, with a 9 choice main menu, and 8 sub menus, also with 9 choices. Actually there are only 8 for patches, as # 9 is the return choice.

I will not try to analyze in depth the almost 700 lines (12 pages printout) long program. Let's just say that it opens random files for the various DOS files, checks and/or changes the pertinent records, and so makes the "patches" without bothering the (slow) PATCH/CMD, or using the, in comparison, snail paced JCL route. And it does it with clever use of DATA statements for menu texts, unique screen manipulation and other slick tricks.

The code is fully accessible, you can study it, and maybe pick some pearls for your own programming. I'm sure Lance wont mind, as long as you give credit where credit belongs. And there are some pearls to be found! So this will be just a description of what the program does, and how it will look to you on the screen. At the end, I'm going to put in a few pennies worth of my own opinion, and hope, Lance will forgive me. I have to restrain myself, not to sound too enthusiastic.

During the program run the two bottom lines are taken out of circulation to display the name, copyright notice, date and real-time clock under a solid line.

The top shows the title, and on the sub menus the section title, and between two lines the status of the 4 areas it handles. That status is updated as soon as you make a patch by just selecting the number from the menu. Even if not prompted on screen, <CTRL><Q> will always go back to the main menu. It's a nice touch to select two keys next to each other. Somewhat easier(?) than the <9><ENTER> combination prescribed in the menu.

Every time you select a sub menu, you will be asked: '...on which drive:', and that drive is then checked for the proper Disk/Command version. If it is not 6.3.1, you will be told so, and advised to use <CTL><Q>, which gives you a chance to select another drive, or change the disk.

If everything is in order, you see the present status of the disk, and can choose to make one or more of the 4 choices to change that status. It checks the patch area, as the ':Fab,cd = yy zz' of the normal PATCH/CMD will do, and if an "unexpected code" is there, will tell so, and that the patch is not made, and again: '--> press Ctrl-Q'.

If ok, astoundingly, no whirr-whirr, blink-blink, but instantaneously, the patch is made. Patching with this (historically slow?) BASIC program is faster then anything I have seen so far.

The ad lists the 32 possible patches (their reversals to normal DOS are implied), but lets elaborate a bit:

The first Sub handles the FORMAT/CMD and lets you disable the Password check/prompt, set up Double Side formatting as default, format 80 tracks as default, and disable the verify. On this last one, I agree with Lance's remark in the manual, that this is not the best of ideas, but some people like to take chances.

The second deals with the BACKUP/CMD: disable the password check, back up Invisible files also as default, you can again disable the verify, and finally override a possible "backup limit", if you have one of those crazy disks.

Next menu deals with DIR and CAT: you can change the commands to a single letter, D or C, and include I files, or both I and S in your DIR and/or CAT as default.

For the PURGE command, the next menu lets you disable the password, include I, or I and S as default, or make Q=N the default setting.

In the DOS COMMAND section, you can change the long REMOVE command to DEL, and shorten RENAME, MEMORY and DEVICE to the first three letters. A boon for a lazy typist like me.

Then comes a menu affecting SYS0 and SYS1: replace the useless ID with the KILL/CMD, enable the BREAK key on AUTO entered with the * parameter, turn the 'SYSGEN' message off at boot up, and maybe most interesting, change, the 'LS-DOS Ready' prompt with your custom message, also 12 characters long. Very valuable to identify a patched DOS. The KILL patch leaves the REMOVE/DEL available, contrary to one patch out there which replaces REMOVE with KILL.

Four MISCELLANEOUS patches are next, affecting SYS0, 2 and 4. Disabling the DATE and/or TIME prompt, disabling the file password protection, and making the extended, 4 line error message default. If you have never seen it, as most "just users" will not have, it tells the error number besides the, sometimes cryptic, message, and gives some SVC and return address info.

And last, but not least, a chance to modify the BOOT. You can chose your very own cursor symbol, anything from ASCII 33 to 191. Some seem more practical then others. Make it solid, boot with the clock on, and change to the fastest key repeat possible.

Needless to repeat that all these custom modifications can be normalized from the same menus. Another nice touch, the odd numbers modify, the even get back to normal. The whole program, as well as the manual, is very well organized.

The 20 page manual leads step by step through the program, explains in every sub-section what prompts are on the screen, what the patches do, what errors could occur and what the error message will say. The BREAK key is disabled during the program to force an orderly exit, and the last manual page gives the reason for that and some short technical info, explaining a few of the tricks.

I tried very hard to trip it up, but not much luck, and nothing serious happened. Yes, it does accept a letter or symbol input at the menu choices, but the following <ENTER> cancels it. And you have to use the <CTL><Q> in sequence, otherwise nothing happens. That catches your lousy typing, and if you **try to** patch a data disk or another DOS then 6.3.1, it tells you either that it is'nt 6.3.1, or that the file (asked for) is not available. Both true, if not specific, depending on what disk you stuck in.

I formatted a data disk with my 6.3.1 FORMAT/CMD, tried all the patch choices, and it told me every time not that the file was'nt there, but that it's not a 6.3.1 disk ??? It gives the same incorrect message for an (unlikely) system disk without these files, contrary to what is said in the manual. It checks the DOS file for 6.3.1 identifying bytes, and if no match, comes to this, sometimes wrong, conclusion, but, surprise, surprise, it puts the filenames FORMAT/CMD, BACKUP/CMD and SYS0, 1, 2, 4, 6 and 7 into the directory, with zero Rec's, EOF, K's and EXT's, and with no date or time. That may be annoying, but not fatal. I dont think this is a reasonable scenario anyway,

But . . . make something fool proof, and a bigger fool comes along. Any comment, Lance?

Now, if it was the '...not 6.3.1' message, and you removed the disk before punching <CTL><Q>, You get a message: '** CLOSE FAULT ** Drive not ready, <ENTER> to retry, <BREAK> to abort'. Dont despair if the <BREAK> wont do anything, it's still disabled. Put the (or another) disk back in and hit <ENTER> (the only active key), it brings the '...which drive:' question back, and <CTL><Q> is alive again. But you wouldn't do a dumb thing like that anyway. None of the above goofy actions do any harm (other than the "extended" directory), nor does it crash the program.

My impressions: A very elegant, flawless working program, with excellent menu screens and with some valuable patches to make the 6.3.1 even more friendly than it already is. A very interesting and innovative handling of a BASIC program code. And a BARGAIN at 15 bucks! I would not know how to evaluate the work and knowledge which went into this program, but it sure is more than I will ever realize. Thank you, Lance.

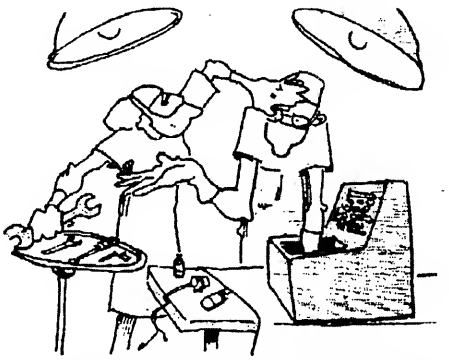
I would have liked the menu choices being Inkeys, and am sorry that the 32 item limit (8 menus with 8 choices each) leaves no way to add patches which may be valuable, such as 24 hr system in DIR, boot in capitals, eliminate leading zeroes at date and time input (a few which I cherish, and which did not interfere at all), and maybe a few others floating around.

Eliminating the verify at FORMAT and BACKUP is not the best policy, just to save a few seconds. Are we that much in a hurry? So is eliminating the DATE and TIME input. It's so easy now, with the period on the number pad, and how does a DIR without date and time look like? Naked!! I am certain that many will find some of these patches of no value to them, and wish some others were included. Maybe Wiz Lance will come up with instructions how to customize his customizing program, suitable for less than super expert BASIC programmers?

Keep the 80's humming!

DR. PATCH REVIEWED

By Chris Fara (Microdex Corp)



This new program, written by Lance Wolstrup, makes all kinds of useful custom changes to LS-DOS version 6.3.1. It includes a 20-page manual and is now available from TRSTimes ($14.95). At this point some people might be tempted to yawn: another patch program? Well, yes and no. Most system patches require typing of obscure commands with strange "syntax" and "hex" codes. It is easy to overlook a "typo" and the result is often disastrous. By contrast, Dr.Patch is entirely menu driven, so even a novice can use it with total confidence.

The menu screens are unusually clean and the whole thing has a neat, almost artistic symmetry. On entry a main menu is presented with 8 sub-menus numbered from 1 to 8. Each sub-menu has also 8 choices. Pick a number and the modification is done. In each menu an odd number changes something in DOS, and the next even number "evens" it back to the normal state. For example, enter 1 to disable passwords, then 2 to re-enable. Once you get the idea, you can practically run the program with closed eyes.

Altogether 32 changes of the system can be instantly done and just as instantly un-done:

- disable password checking and the bothersome "backup limit" schemes;
- set up custom "defaults" for format, backup, purge and directory;
- shorten frequently used commands (such as DIR to D, REMOVE to DEL, etc);
- streamline computer start-up sequence;

and so on, too much to list it all.

My immediate favorite became the quick change of the dull "DOS Ready" prompt to whatever, for instance "Hello Chris!". Actually this is not just a cute gimmick. Even if you apply patches other than from Dr.Patch, it is not a bad idea to change the prompt to signal a customized system.

A tantalizing proposition are patches that eliminate the "verify" in format and backup, and thus dramatically speed up the process. The manual warns that this should be done only with top-quality disks. Maybe so, but I've been always wondering. Having gone through thousands of disks, many of the cheap bulk variety, I found precious few that wouldn't "verify". On the other hand, I also ran into brand-name disks with a "lifetime warranty", that "verified" perfectly, yet files got messed up anyway. So it seems that in routine work, assuming properly aligned disk drives, the risk is minimal, and the snappy format and backup are a pleasure.

Now for the clincher: the whole program is written in straight BASIC. What? Can't do! The "real programming" is supposed to be in assembly language, or better yet, in C, C+, C++, C+++++++ and such. This CW (Conventional Wisdom) is what apparently prompted the author to demonstrate that yes, BASIC can do just fine. And indeed it does.

As with the screens, the artistic quality is also visible in the code listing which is not "protected". Well organized, easy to read, it has a great educational value. With some knowledge of BASIC, one can even customize the program itself. Prefer ERASE instead of REMOVE or DEL to delete files? Just change a few keywords in the listing.

Being a programmer myself, I am obviously biased, so take my few gripes here with a grain of salt. One thing I found really awkward was that before displaying any sub-menu, the program first requires selection of the drive to be patched, and then goes through checking of that drive. I'd rather see the menu first. It might not have the patches I am looking for, in which case disk selection and checking is a waste of time.

The program takes up a hefty 22.5-K of the disk space. Since all 64 routines (32 patches and 32 un-patches) follow nearly identical steps, a subroutine scheme could probably cut the program size in half without sacrificing any of its remarkable speed (but, as the author replied in defense, this would reduce the legibility of the code). The manual recommends copying the program to the system disk. But, because of its size, a more practical approach is to BACKUP BASIC(INV) to the program disk. Label it "Dr.Patch", keep it handy, and stick it into any data drive whenever a patch is desired.

And I really mean: keep it handy. Never mind my habitual fussing about details. As it stands, the program is useful, bug-free and, unlike some other system patches, completely harmless. Have fun.

*** DR. PATCH ***

BRAND-NEW UTILITY FOR TRS-80 MODEL 4 AND LS-DOS 6.3.1

A 'MUST HAVE' FOR ALL LS-DOS 6.3.1 OWNERS.

DR. PATCH MODIFIES LS-DOS 6.3.1 TO DO THINGS THAT WERE NEVER BEFORE POSSIBLE.
COMPLETELY SELF-CONTAINED - MENU-DRIVEN FOR MAXIMUM USER CONVENIENCE.
FAST & SAFE - EACH MODIFICATION IS EASILY REVERSED TO NORMAL DOS OPERATION.

DISABLE PASSWORD CHECK IN FORMAT/CMD
FORMAT DOUBLE-SIDED AS DEFAULT
FORMAT 80 TRACKS AS DEFAULT
DISABLE VERIFY AFTER FORMAT
CHANGE 'DIR' TO 'D'
CHANGE 'CAT' TO 'C'
VIEW DIR/CAT WITH (I) PARAMETER AS DEFAULT
VIEW DIR/CAT WITH (S,I) PARAMETERS AS DEFAULT
CHANGE 'REMOVE' TO 'DEL'
CHANGE 'RENAME' TO 'REN'
CHANGE 'MEMORY' TO 'MEM'
CHANGE 'DEVICE' TO 'DEV'
DISABLE THE BOOT 'DATE' PROMPT
DISABLE THE BOOT 'TIME' PROMPT
DISABLE FILE PASSWORD PROTECTION
ENABLE EXTENDED ERROR MESSAGES

DISABLE PASSWORD CHECK IN BACKUP/CMD
BACKUP WITH (I) PARAMETER AS DEFAULT
BACKUP WITH VERIFY DISABLED
DISABLE BACKUP 'LIMIT' PROTECTION
DISABLE PASSWORD CHECK IN PURGE
PURGE WITH (I) PARAMETER AS DEFAULT
PURGE WITH (S,I) PARAMETERS AS DEFAULT
PURGE WITH (Q=N) PARAMETER AS DEFAULT
IMPLEMENT THE DOS 'KILL' COMMAND
CHANGE DOS PROMPT TO CUSTOM PROMPT
TURN 'AUTO BREAK DISABLE' OFF
TURN 'SYSGEN' MESSAGE OFF
BOOT WITH NON-BLINKING CURSOR
BOOT WITH CUSTOM CURSOR
BOOT WITH CLOCK ON
BOOT WITH FAST KEY-REPEAT

DR. PATCH IS THE ONLY PROGRAM OF ITS TYPE EVER WRITTEN FOR THE TRS-80 MODEL 4 AND LS-DOS 6.3.1. IT IS DISTRIBUTED EXCLUSIVELY BY TRSTIMES MAGAZINE ON A STANDARD LS-DOS 6.3.1 DATA DISKETTE, ALONG WITH WRITTEN DOCUMENTATION.

*** DR. PATCH *** $14.95

SHIPPING & HANDLING: U.S & CANADA - NONE
ELSEWHERE - ADD $4.00
(U.S CURRENCY ONLY, PLEASE)

TRSTimes magazine - dept. DP
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

DON'T LET YOUR LS-DOS 6.3.1 BE WITHOUT IT!

CITIZEN SETUP

Model 4 - LS-DOS 6.x.x - Editor/Assembler

By M.C Matthews

PRINTSET is an assembly language program that allows easy setup of a Citizen 120D printer directly from DOS, or from Basic, if desired. When executed, PRINTSET gives the following choices:

10 or 12 Characters/inch (1/2)
Normal or compressed print (N/C)
Single or Emphasised (S/E)
Paper length (ENTER) for 66
Bottom margin (ENTER for none,
or enter number of lines)
Slashed 00s (Y/N)

Get out your editor/assembler, type in the listing below and assemble it as PRINTSET/CMD. The program should work on most Citizen printers.

```
; PRINTSET/CMD TO SET PRINTER FROM DOS.
          ORG   2400H
CLS       EQU   0545H
START     CALL  CLS
          LD    HL,0000H
          LD    A,0FH
          LD    B,03H
          RST   28H
          DEC   HL
NEXT1     LD    HL,MESS1      ;10/12 chars/in.
          LD    A,0AH
          RST   28H           ;Display it
TRY1      LD    A,1
          RST   28H           ;Get answer
          LD    C,A
          CALL  SHOWIT        ;Display it
          CP    31H           ;1 for 10 chars
          JR    NZ,TRY1A
          LD    A,50H         ;Change so put new in
          LD    (MESS4+8),A
          JR    NEXT2
TRY1A     CP    32H           ;2 for 12 chars
          JR    NZ,NEXT1      ;12 chars normal
NEXT2     LD    HL,MESS2      ;Normal/Compressed
          LD    A,0AH
          RST   28H
TRY2      LD    A,1
          RST   28H
          LD    C,A
          CALL  SHOWIT
          CP    4EH           ;Normal, no change
          JR    Z,NEXT3
          CP    6EH
          JR    Z,NEXT3
TRY2B     CP    43H           ;Compressed
          JR    Z,PUT2B
          CP    63H
          JR    NZ,NEXT2
PUT2B     LD    A,0FH         ;Compressed chars
PUT2      LD    (MESS4+9),A   ;Put in place
NEXT3     LD    HL,MESS3      ;Single or emphasised
          LD    A,0AH
          RST   28H
TRY3      LD    A,1
          RST   28H
          LD    C,A
          CALL  SHOWIT
          CP    53H           ;Single
          JR    Z,NEXT4
          CP    73H
          JR    Z,NEXT4
          CP    45H           ;Emphasised
          JR    Z,PUT3
          CP    65H
          JR    NZ,NEXT3
PUT3      LD    A,45H
          LD    (MESS4+11),A
NEXT4     LD    HL,MESS6      ;Paper length
          LD    A,0AH
          RST   28H
          LD    A,09H
          LD    HL,BUFF       ;Set 3-byte buffer
          LD    B,02H
          RST   28H           ;Get number
          JR    NZ,NEXT4
          LD    A,60H
          RST   28H           ;Convert from ASC to Hex
          LD    A,C           ;Get lower byte of answer
          CP    00H
          JR    Z,NEXT5       ;Nul so 66 lines
          LD    (MESS4+14),A  ;Put in place
NEXT5     LD    HL,MESS7      ;Margin at foot
          LD    A,0AH
          RST   28H
          LD    HL,BUFF
          LD    B,02H
          LD    A,09H
          RST   28H
          LD    A,60H
          RST   28H
          LD    A,C
          CP    00H
          JR    Z,NEXT6
TRY5A     LD    A,C
          LD    (MESS4+21),A  ;Set length of margin
          LD    A,4EH         ;Set for margin
          LD    (MESS4+20),A
NEXT6     LD    HL,MESS8      ;Slashed 0s?
          LD    A,0AH
          RST   28H
TRY6A     LD    A,1
          RST   28H
          LD    C,A
          CALL  SHOWIT
          LD    A,C
```

```
         CP    4EH          ;No
         JR    Z,DONE
         CP    6EH          ;no
         JR    Z,DONE
         CP    59H          ;Yes
         JR    Z,PUT6
         CP    79H          ;yes
         JR    NZ,NEXT6
PUT6     LD    A,01H        ;1 for slashed
         LD    (MESS4+18),A
DONE     LD    C,0AH        ;Do line feed
         CALL  SHOWIT
         CALL  DOPRT        ;See if printer ready
         LD    HL,MESS4     ;Send instructions
         LD    A,0EH
         RST   28H          ;to printer
         LD    HL,0000H
         RET                ;Exit to DOS
SHOWIT   LD    A,02H
         RST   28H          ;Display a char.
         RET
MESS1    DEFM 'PRINTSET. Set up Citizen 120D printer'
         DEFB 0AH
         DEFM 'Vers.15.  25.04.91  M.C.Matthews'
         DEFB 0AH
         DEFB 0AH
         DEFM '10 or 12 Characters/inch?  (1/2) '
         DEFB 03H
MESS2    DEFB 0AH
         DEFM 'Normal or compressed print? (N/C) '
         DEFB 03H
MESS3    DEFB 0AH
         DEFM 'Single or emphasised? ?  (S/E) '
         DEFB 03H
MESS4    DEFB 1BH
         DEFB 7EH           ;Set Epson
         DEFB 35H
         DEFB 00H
         DEFB 1BH           ;Set English chars.
         DEFB 52H
         DEFB 33H
         DEFB 1BH
         DEFB 4DH           ;12 chars/in.
         DEFB 12H           ;Normal type
         DEFB 1BH
         DEFB 48H           ;Single struck
         DEFB 1BH
         DEFB 43H
         DEFB 42H           ;Paper length
         DEFB 1BH
         DEFB 7EH
         DEFB 34H
         DEFB 00H           ;No slashed 0
         DEFB 1BH
         DEFB 4FH
         DEFB 00H           ;Margin
         DEFB 0DH
DOPRT    CALL  SETPTR       ;Check printer ready
         LD    HL,MESS4
         LD    A,0EH
         RST   28H          ;Send instructions
         LD    HL,0000H
         RET                ;Back to where we came from
SETPTR   LD    DE,5250H     ;Get SVC location
         LD    A,52H
         RST   28H
         EX    DE,HL
         LD    C,00H
         LD    A,05H
         RST   28H          ;Test for ptr.ready
         JR    Z,READY
         LD    HL,1700H     ;Else set cursor to
         LD    B,03H        ;bottom of screen
         LD    A,0FH
         RST   28H
         LD    HL,MESS5
         LD    A,0AH        ;& display message
         RST   28H
         LD    B,80H        ;Delay
         LD    DE,-1
LOOPA    LD    HL,0FFH
LOOPB    ADD   HL,DE
         JR    C,LOOPB
         DJNZ  LOOPA
         LD    HL,1700H     ;Reset cursor
         LD    B,03H
         LD    A,0FH
         RST   28H
         LD    HL,MESS9     ;& blank out message
         LD    A,0AH
         RST   28H
         LD    B,80H        ;Another delay
         LD    DE,-1
LOOPA1   LD    HL,0FFH
LOOPB1   ADD   HL,DE
         JR    C,LOOPB1
         DJNZ  LOOPA1
         JR    SETPTR       ;& back for another try
READY    RET
BUFF     DEFB 00H
         DEFB 00H
         DEFB 00H
MESS5    DEFM '* * PRINTER OFF LINE * *'
         DEFB 03H
MESS6    DEFB 0AH
         DEFM 'Paper length? (ENTER for 66) '
         DEFB 03H
MESS7    DEFM 'Bottom margin? (ENTER for none,'
         DEFB 0AH
         DEFM 'or enter number of lines) '
         DEFB 03H
MESS8    DEFM 'Slashed 00s? (Y/N) '
         DEFB 03H
MESS9    DEFM '                   '
         DEFB 03H
         END  START
```

FLOPPY DISK READ/WRITE TECHNOLOGY.

By Karl Mohr

Magnetic materials have figured heavily in the spectacular growth of the electronics industry. The reason is that the permanent method for storing information in computers- ranging from personal computers to large mainframes- is magnetic recording. In one form of the technology a rigid 14-inch aluminum disk coated with iron oxide is spun about its axis at 3,000 revolutions per minute. Such a rate of spin corresponds to a speed of more than 100 miles per hour at the edge of the disk. As the disk rotates, a 'head' with which to 'read' and 'write' data is brought near it. The head consists of a coil of wire wrapped around a magnetic core, which is typically a nickel-iron alloy.

By passing an electric current through the head coil one can record data on the disk. The current generates a magnetic field in the coil, which magnetizes a particular area of iron oxide on the disk. That area retains its magnetization and so can 'remember' information. The process by which data are encoded on the disk is known as the write cycle. Information on the disk is read out with the same head by reversing the procedure. As the head moves over the disk, magnetized regions in the disk induce a current in the coil. By measuring the current as a function of time the stored information is obtained.

Even though the process of reading and writing are based on simple principles, attempts to pack more information onto disks have proved somewhat problematical. As the bit density increases, the magnetic field of each bit decreases. In order to detect the smaller magnetic field, the head must be positioned closer to the disk. Unfortunately smaller head-to-disk spacings increase the probability of occasional but devastating impacts of the head onto the disk. Moreover, since the surface of the magnetic layer is usually quite rough-on the order of 1,000 angstroms-frictional wear is quite common.

To minimize damage from both sudden impacts and frictional wear, lubricants are applied to the disk. Of course, frictional wear could also be reduced if the surface of the disk were made smoother. To build smoother surfaces manufacturers of disk memories are exploring the possibility of exploiting thin-film deposits techniques, in which layers of material are deposited by a plating or evaporation technique. In spite of the successful history of inductive recording, however, the tribology of disks is still more an art than it is a science.

In order to build heads that are more sensitive to magnetic fields, investigators have recently begun to borrow materials and techniques from semiconducting-chip technology. IBM, for example, has introduced a head in which the traditional coil of wire has been replaced by a thin-film conductor deposited as a spiral on the surface of a silicon substrate. The magnetic core of the head is Permalloy, a mixture of nickle and iron. The head, which is now part of the IBM Model 3370 disk memory, can respond to variations in current that occur as rapidly as 100 MILLION times per second. In addition the magnetic field delivered by the head is capable of recording 15,000 bits of information per inch!

In the long run it would be advantageous to eliminate the head altogether. One approach involves the use of magneto-optic materials: magnetic materials that can affect the properties of light. A number of different magneto-optic materials have been investigated over the past two decades. The ones currently in favor are amorphous-alloy films containing rare earth elements such as Gadolinium and Terbium and transition metals such as iron and cobalt. These ideas were first proposed in the early 1970's by persons working at IBM. Investigators in Japan, Europe and most recently in the U.S. have since contributed to the development of the materials and their use in the fabrication of prototype storage devices.

Information is written onto a magneto-optic disk by simultaneously applying a magnetic field and a pulse of laser light to a spot on the disk. The laser light heats the spot while the applied magnetic field magnetizes it. The direction of magnetization of the spot is determined by the direction of the applied magnetic field. To read the data, a beam of polarized light is shone on the disk. The polarization of the reflected light is changed according to the direction of magnetization at each point on the disk. By measuring the change in polarization, one can access the stored information.

Disk memories based on all-optical elements are also under development, that however is another topic!

Model I & III
Public Domain Disks

PD#1: binclock/cmd, binclock/doc, checker/bas, checker/doc, chomper/bas, cls/cmd, dduty3/cmd, driver/cmd, driver/doc, drivtime/cmd, mazeswp/bas, minibase/bas, minitest/dat, mx/cmd, piazza/bas, spdup/cmd, spdwn/cmd, vici/bas, vid80/cmd, words/dic.

PD#2: creator/bas, editor/cmd, maze3d/cmd, miner/cmd, note/cmd, poker/bas, psycho/cmd, supdraw/cmd, vader/cmd

PD#3: d/cmd, trsvoice/cmd, xmodem/cmd, xt3/cmd, xt3/txt, xthelp/dat

PD#4: cobra/cmd, disklog/cmd, flight/bas, flight/doc, narzabur/bas, narzabur/dat, narzabur/his, narzabur/txt, othello/bas, vid80x24/cmd, vid80x24/txt

PD#5: eliza/cmd, lu31/cmd, sq31/cmd, usq31/cmd

PD#6: clawdos/cmd, clawdos/doc, cocoxf40/cmd, dskrnam/bas, menu/cmd, ripper3/bas, sky2/bas, sky2/his, space/cmd, stocks/bas, trs13pat/bas, vidsheet/bas

PD#7: cards/bas, cities/bas, coder/bas, eye/bas, heataudt/bas, hicalc/bas, life/bas, moustrap/bas, ohare/bas, slots/bas, stars/cmd, tapedit/bas

PD#8: craps/bas, fighter/bas, float/bas, hangman/bas, jewels/cmd, lifespan/bas, varidump/bas, xindex/bas, xor/bas

PD#9: bublsort/bas, chess/bas, finratio/bas, homebudg/bas, inflat/bas, mathdril/bas, midway/bas, nitefly/bas, pokrpete/bas, teaser/bas

PD#10: ltc21/bas, ltc21/ins, lynched/bas, match/bas, math/bas, message/bas, message/ins, portfol/bas, portfol/ins, spellegg/bas, storybld/bas

PD#11: alpha/bas, caterpil/cmd, cointoss/bas, crolon/bas, cube/cmd, dragon/cmd, fastgraf/bas, fastgraf/ins, lunarexp/bas, music/bas, music/ins, planets/bas, volcano/cmd

PD#12: baccarat/bas, backpack/bas, backpack/ins, doodle/bas, dragons/bas, dragons/ins, king/bas, sinewave/bas, snoopy/bas, wallst/bas, wallst/ins

PD#13: atomtabl/bas, boa/bas, chekbook/bas, conquer/cmd, dominos/bas, morse/bas, mountain/bas, quiz/bas, signbord/bas, sketcher/bas

PD#14: autoscan/bas, checkers/bas, craps/bas, ducks/bas, isleadv/bas, nim/bas, rtriangl/bas, sammy/cmd, typing/bas, wordpuzl/bas

PD#15: budget/bas, corp/bas, corp/ins, fourcolr/bas, fullback/bas, grapher/bas, illusion/bas, jukebox/bas, ledger/bas, maze/cmd, reactest/bas, shpspree/bas, states/bas, tapecntr/bas, tiar/bas, tiar/ins

PD#16: amchase/bas, constell/bas, filemastr/bas, foneword/bas, geometry/bas, heartalk/bas, hidnumbr/bas, lgame/bas, marvello/bas, powers/bas, scramble/bas, speed/bas, subs/bas

PD#17: conundrm/bas, eclipse/bas, esp/bas, esp/ins, hustle/bas, jacklant/bas, mindblow/bas, othello/bas, pleng/bas, rubik/bas, trend/bas, ufo/bas, veggies/bas

PD#18: backgam/bas, chess/cmd, cosmip/cmd, distance/bas, hexpawn/bas, music/cmd, stokpage/bas, texted/bas, texted/ins, trex/bas, twodates/bas, wanderer/bas

PD#19: banner/bas, cresta/cmd, lander/bas, medical/bas, moons/bas, par/bas, parchut/bas, pillbox/bas, readtrn/bas, replace/bas, ship/cmd, solomadv/bas, space/cmd, survival/bas

PD#20: bomber/bas, bumbee/cmd, ciaadv/bas, dice31/bas, dice31/ins, diskcat1/bas, firesafe/bas, flashcrd/bas, hitnmiss/bas, mazegen/bas, mazescap/cmd, roulette/bas, seasonal/bas

PD#21: aprfool/bas, catmouse/bas, d/cmd, escape/bas, header/bas, kalah/bas, mathwrld/bas, nameit/bas, note/cmd, photo/bas, read/cmd, syzygy/bas, timeshar/cmd, timeshar/doc, trace80/cmd, trsdir/cmd, worm/bas, yatz80/bas

PD#22: arcade/bas, cube/cmd, eclipse/bas, lcd/bas, leastsqr/bas, medical/bas, million/bas, pwrplant/bas, round/bas, subway/bas, tapeid/bas

PD#23: artil/bas, artil/ins, baseconv/bas, crushman/bas, dissert/bas, huntpeck/bas, jungle/bas, jungle/ins, messages/bas, monitor/bas, monster/bas, moons/bas, ohmlaw/bas, stockpage/bas, tictacto/bas

PD#24: baslist/asm, baslist/cmd, baslist/doc, cleaner3/cmd, cleaner3/doc, difkit1/bas, difkit1/doc, dirpatch/asm, dirpatch/cmd, e/cmd, ei/doc, i/cmd, newmap/bas, newmap/doc, varlst/asm, varlst/cmd, varlst/doc

PD#25: copy/bas, copy/doc, dirpw/asm, dirpw/cmd, dirpw/doc, dskfmt/bas, dskfmt/doc, himap/asm, himap/cmd, huricane/bas, hv/bas, hv/doc, keydemo/bas, keyin/bas, keyin/doc, lazyptch/asm, lazyptch/doc, salvage/bas, salvage/doc,wpflt/asm, wpflt/flt

PD#26: constell/bas, divisor/bas, frame/bas, heatfus/bas, heatfus/doc, hicalc/bas, mathlprt/bas, mathquiz/bas, molecule/bas, morscode/bas, phyalpha/bas, phyalpha/doc, remaindr/bas, usa/bas, wiring/bas

PD#27: engine/bas, fraction/bas, geosat/bas, grades/bas, julian/bas, lunarcal/bas, mailist/bas, metaboli/bas,musictrn/bas, perindex/bas, potrack/bas

PD#28: chainfil/bas, citoset/bas, convnum/bas, cursors/bas, cursors/doc, datamkr/bas, deprec/bas, gmenuii/bas, ledger12/bas, menui/bas, menuii/bas, minives/bas, ninteres/bas, refinanc/bas, regdepo/bas, rembal/bas, rndbordr/bas

Each disk is $5.00 (U.S.)
or get any 3 disks for $12.00 (U.S.)

please specify the exact disks wanted.

TRSTimes PD-DISKS
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367

NOTES ON VISICALC

By Jim E. King

In order to track of the gas mileage of my 1971 VW Van, I wrote a template for Model 4 Visicalc, apporpriately called VAN/VC; the listing can be found below.

Rows 1 through 3 are text. Line 1 tells what this sheet is; it has a place for the complete beginning odometer reading. Line 2 shows that the first 4 columns are your input: number of gallons, Dollar cost, and just for a check, the price per gallon showing on the pump.

The instructions for calculating car mileage start with recording the odometer reading at a full tank. That number is at A4 in the sheet, and is used in many of the equations. The xx-- in row 3 are there merely because no calculation is possible on only that initial mileage entry.

When you move the cursor to the input boxes you will note that the same numbers appear at the top of the screen. These are the numbers that you input.

When you move the cursor into the boxes under Total or Delta, equations appear at the top.

In box E5 under Total Miles you will see +A5-A4 at the top. This means that the program takes the VALUE in box A5 and then subtracts from it the VALUE in A4 and displays the answer in box E5. These variables are other box locations. As you move the cursor down column E you will note that the A# value increases, but A4 stays the same; this is the Total Miles traveled to that point.

Total Gallons is a little different. F5 says +B5, the first fill-up. F6, however says F5 + B6 = the total in the previous F box plus the new. F7 says F6 + B7.

Total $ is calculated exactly like Gallons.

Total Miles/Gallon is a simple division. H5 = E5/F5, etc.

Total Cents/Mile is similar: I5 = 100*G5/E, etc.

Delta means NOT total, just the ratios for only that portion of the data. Delta Miles, J5 = A5-A4. J6 = A6-A5, etc.

Delta Miles/Gallon, K5 = J5/B5, etc.

Delta Cents/Mile, L5 = 100*C5/J5, etc.

Calculated $/Gallon, M5 = C5/B5, etc.

VAN/VC

```
M14:@INT(C14/B14*10000+.5)/10000
L14:+A14-A13
K14:/F$+C14*100/L14
J14:/F$+L14/B14
I14:/F$+G14*100/E14
H14:/F$+E14/F14
G14:+G13+C14
F14:+F13+B14
E14:+A14-A4
D14:.789
C14:8.34
B14:10.57
A14:2154.1
M13:@INT(C13/B13*10000+.5)/10000
L13:+A13-A12
K13:/F$+C13*100/L13
J13:/F$+L13/B13
I13:/F$+G13*100/E13
H13:/F$+E13/F13
G13:+G12+C13
F13:+F12+B13
E13:+A13-A4
D13:.809
C13:7.42
B13:9.17
A13:1951.6
M12:@INT(C12/B12*10000+.5)/10000
L12:+A12-A11
K12:/F$+C12*100/L12
J12:/F$+L12/B12
I12:/F$+G12*100/E12
H12:/F$+E12/F12
G12:+G11+C12
F12:+F11+B12
E12:+A12-A4
D12:.869
C12:9.36
B12:10.77
A12:1802.7
M11:@INT(C11/B11*10000+.5)/10000
L11:+A11-A10
K11:/F$+C11*100/L11
J11:/F$+L11/B11
I11:/F$+G11*100/E11
H11:/F$+E11/F11
G11:+G10+C11
F11:+F10+B11
E11:+A11-A4
D11:.879
C11:/F$10
B11:11.38
A11:1640
M10:@INT(C10/B10*10000+.5)/10000
L10:+A10-A9
K10:/F$+C10*100/L10
J10:/F$+L10/B10
I10:/F$+G10*100/E10
H10:/F$+E10/F10
G10:+G9+C10
F10:+F9+B10
E10:+A10-A4
D10:.899
C10:/F$7.5
B10:8.34
A10:1387.7
M9:@INT(C9/B9*10000+.5)/10000
L9:+A9-A8
K9:/F$+C9*100/L9
```

```
J9:/F$ + L9/B9
I9:/F$ + G9*100/E9
H9:/F$ + E9/F9
G9: + G8 + C9
F9: + F8 + B9
E9: + A9-A4
D9:.889
C9:/F$10
B9:11.25
A9:1250
M8:@INT(C8/B8*10000 + .5)/10000
L8: + A8-A7
K8:/F$ + C8*100/L8
J8:/F$ + L8/B8
I8:/F$ + G8*100/E8
H8:/F$ + E8/F8
G8: + G7 + C8
F8:/F$ + F7 + B8
E8: + A8-A4
D8:.909
C8:9.14
B8:10.05
A8:1043.7
M7:@INT(C7/B7*10000 + .5)/10000
L7: + A7-A6
K7:/F$ + C7*100/L7
J7:/F$ + L7/B7
I7:/F$ + G7*100/E7
H7:/F$ + E7/F7
G7: + G6 + C7
F7: + F6 + B7
E7: + A7-A4
D7:.929
C7:/F$11.34
B7:12.21
A7:870.4
M6:@INT(C6/B6*10000 + .5)/10000
L6: + A6-A5
K6:/F$ + C6*100/L6
J6:/F$ + L6/B6
I6:/F$ + G6*100/E6
H6:/F$ + E6/F6
G6: + C5 + C6
F6: + B5 + B6
E6: + A6-A4
D6:.949
C6:/F$7
B6:7.38
A6:663
M5:@INT(C5/B5*10000 + .5)/10000
L5: + A5-A4
K5:/F$ + C5*100/L5
J5:/F$ + L5/B5
I5:/F$ + G5*100/E5
H5:/F$ + E5/F5
G5: + C5
F5: + B5
E5: + A5-A4
D5:.959
C5:/F$11.61
B5:11.96
A5:530
M4:" xxxxxxxx
L4:" xxxxxxxx
K4:" xxxxxxxx
J4:" xxxxxxxx
I4:" xxxxxxxx
H4:" xxxxxxxx
G4:" xxxxxxxx
F4:" xxxxxxxx
E4:" xxxxxxxx
D4:" xxxxxxxx
C4:" xxxxxxxx
B4:" xxxxxxxx
A4:314.2
M3:/FR"$/Gal
L3:/FR"Miles
K3:/FR"Cts/Mi
J3:"  Mi/Gal.
I3:/FR"Cts/Mi
H3:/FR"Mi/Gal
G3:"    $
F3:/FR"Gals
E3:/FR"Miles
D3:/FR"$/Gal
C3:"    $
B3:/FR"#Gals
A3:"Odometr
M2:/FR"Calc
L2:/FR"Delta
K2:/FR"Delta
J2:/FR"Delta
I2:/FR"Total
H2:/FR"Total
G2:/FR"Total
F2:/FR"Total
E2:/FR"Total
D2:/FR"Input
C2:/FR"Input
B2:/FR"Input
A2:/FR"Input
M1:/FR"Van
K1:20314.2
J1:"eter:
I1:"ng Odom
H1:"Beginni
F1:"Van
E1:"ions:
D1:"alculat
C1:"LEAGE C
B1:/FR"Gas MI
/W1
/GOR
/GRA
/GC7
/XA1:A1:
```

DIRECT from CHRIS

The shortest distance between YOU and DOS

Review by Allen Jacobs

Interacting with DOS has always been a problem with every computer system ever made. The flexibility we have in our choices of DOS interaction methods depends upon the computer hardware we are dealing with. User interfaces can range from an on-off switch or gear lever in the earliest systems to enjoying Star Trek's Holideck or pondering the personality of Mr. Data, himself. The latter two are futuristic examples of advanced user interfaces. Even The Federation has only been able to afford to produce one Mr. Data. The obvious trend we see is that the greater and faster the equipment available to the system is, the more complex, information rich, feature filled, and elegant the user interface that one can have, may be.

In the TRS-80, our system resources put practical limits on our user interface. But don't feel left out by the capacity of our machines. I hear that Windows applications run too slowly on all but the fastest '486 machines. This is true even when they are equipped with special speed-up cards required just for the complex graphical user interface, when it is in high resolution video mode. My sympathies. It seems that a corollary to the rule that programs will expand to fill all available memory is that the program that takes up the most room will be the user interface.

All that interfaces do is to provide us with information about the system and allow us to control it with some form of instructions. The tradeoff between interfaces in a given system, such as ours, is that the more you have the computer do in the form of providing information and interpreting the user response, the greater the portion of system resources you will require. If you ask too much of it, you slow it down and occupy system storage capacity. Therefore, an example of overloading even a high capacity system is expecting Mr. Data to understand and laugh at one of Captain Picard's obscure jokes. Instead, his response time slows, he struggles, and eventually he gives some diplomatic excuse, explaining why he can not discern the humor in the statement. That's just a fancy way of Mr. Data issuing a system overload error message through his user interface, his personality.

We can see that skillful allocation of system resources to the user interface is partially art and partially personal preference. For the TRS-80, the range of interfaces available to us starts at the DOS READY prompt and ends with Deskmate. DOS READY is fast and takes up no additional program space, but you have to know what your choice of commands is and you have to type them into the machine in an error free manner, repeatedly. If not, you will find yourself swearing at the DOS ERROR and PROGRAM NOT FOUND messages you will often encounter. All information about the system, including the time of day must specifically be requested. This form of interface is easy on the DOS but hard on you.

At the other end of things, with Deskmate, you can have a DOS interface that takes up 384K of space on a hard disk (without ARC, PDS, or diskDISK), provides an ever present calendar, and requires that you jump between numerous "windows" on the screen until you get to the one you want. The only programs I can think of that are more cumbersome are Multiplan for the TRS-80 and Prodigy for MS-DOS.

Now that we have established a continuum of the solutions for efficiently interfacing with DOS, that finally brings us to a nice middle of the road DOS interface program called "DIRECT", by Chris, from Microdex. In its minimal configuration, it requires as little as 3K of disk space. This means that it can reside on a "fresh", unaltered LS-DOS 6.3.1 system disk, a single sided floppy. In memory, it resides in the DOS overlay area so it never interferes with "normal" DOS programs. That feature alone can "save-your-bacon" after a system crash. Its instructions are comprehensive but can be summarized quite simply.

DIRECT installs itself as the ECI (Extended Command Interpreter) which can optionally be made active so that DIRECT substitutes for the DOS READY prompt. It can also be activated by entering an <*> at DOS READY.

DIRECT is not a 7K-14K+ point-and-shoot menu system such as Shell or Deskmate. They work by printing the entire catalog of a given drive to the screen while highlighting one file at a time, directable by the arrow keys. When the desired program is selected, another key press executes the program name as if it was typed onto the command line.

Instead, DIRECT is a customizable command line editor that can store up to 58 different lines of DOS commands in menu assisted files and can transfer to any of the total of 36 possible menus (ie.: 0-9, and A-Z). Each of these can contain up to 58 command lines of its own.

Potentially, it can require less keystrokes to call a program with DIRECT than with a menu program such as Shell or Deskmate. These programs can require a number of arrow key presses to get to your program if it is at the bottom-right of a large CAT listing. DIRECT can be edited to get to the same file with the press of ONE key.

The menus, which are the entire screen except for the bottom line, have no automatic control over DOS. Rather, they are user composed information screens that remind the operator of the commands available within the menu and the keys required to activate each line. Importantly, command lines an any menu can call other menus, all at the discretion of the user.

Both the menus and the command lines are independent of each other. They are editable in a free-form man-

ner. Command line users will find this capability a dream-come-true. They will say, "Finally, I have a place to make notes and to take the time to edit that long command line I always hate to compose because of all the #$%& switches I have to set so I can't use a full menuing program."

Full menu program users will find themselves hitting the enter key and wondering what is happening. They will wait for a prefabricated screen to come up. When it does, it will ask that they compose this screen in any way they desire, providing information and notes about their programs and/or DOS. At this point, a percent of the users will probably freeze, thinking that they have done something wrong. Only a lack of curiosity will insulate them from discovering the utility of the program from that point onward. Any slight curiosity and experimentation, however, will reward them with a true user's convenience.

The editor that comes with the program resides in another file that may be kept on another disk, along with any other user composed menus. Only menu "0" is required to be present to the system. The editor is not; but since it is callable from the main program, it has to be available somewhere.

The instructions are as well written as any can be for a program of this nature. Frankly, however, they will hardly make any sense until you actually try to run the program. By the time you have figured nearly everything out, which takes about an hour of devoted playing around, you will only need the instructions to determine a few syntax rules. These, you will immediately write to a menu to remind you of them. As you use the program more, you will find that you remember them anyway because you use them all the time. After a while, you will forget that the program is not just part of DOS and you will include it on all of your system floppies. If you use double sided drives you will find it more useful than some seldom used system utilities. In fact, you would consider killing them, instead of DIRECT, if you were short of space.

The only three really important differences between writing a command line into DIRECT and DOS are the additional uses of punctuation. They are as follow:

Writing an asterisk at the beginning of a line, followed by a letter or number, calls that menu as the current screen. If it doesn't exist, you are asked if you want to create it. The editor must be present to do that.

If you follow a command line with one or more periods, DIRECT will pause and prompt you to continue by pressing any key.

If you follow the command line with a question mark, you may add the name of a particular file or any other command parameters from the keyboard, before hitting the enter key to execute the command.

That's about all you have to know to get started with the program. Pretty simple, isn't it?

While Chris says that DIRECT works best on a hard drive system, I disagree to an extent. A large hard drive has enough space for a full menu driven program whereas a "Memdisk" or "Grafdisk" system drive strikes me as the ideal place for DIRECT. Here, it is small enough to fit where a 14K "Shell" or a who-knows-how-many-K "Deskmate" just won't go. Yet, it can be configured to provide all the functionality of Shell while being tightly customized for the floppy application being run. In this context, I can think of no more ideally suited program. THAT also happens to be the way the majority of Model 4 systems are operated, most of the time.

An interesting use for DIRECT is a quick way to create a prefabricated menu system for another user. It will be effective, but it will not be bullet proof. This is because pressing <F2> will always get you to DOS READY. However, by eliminating the editor, the choices you have placed into a menu cannot easily be altered by a causal user.

I do have some minor criticisms about the program. It seems that it takes too many keystrokes to get out of the editor when you finish editing a menu. It appears to me that I am gratuitously requested to save the menu twice. I am not sure why I am asked to do so, but there may be a situation when such an action is appropriate. At this time, however, I can not see why.

Also, on trying to create each of the graphic characters while attempting to produce a listing of all them, I discovered that you can't create every one. This problem may occasionally prove to be a program limitation, but it appears to be acceptable as long as you can do boxes.

When you begin editing from a given menu, the editor asks you what menu you would like to edit immediately after it has eliminated the number of the current menu from the screen. Also, the request itself does not default to the current screen. Therefore, if you don't remember which screen you want to edit, you must guess and break out of the procedure later, if you were wrong. This is regrettable since the name of the menu is retained and displayed as the default menu, AFTER you are done editing it.

What is especially nice about DIRECT, however, is that when the machine is left alone, a screen saver blanks the screen, with the optional exception of the system clock. Depressing any key restores the current screen. It can also be made to beep when left alone, but I don't know that I have a particular need for that feature. Also, DOS is only that one key, <F2>, away from any part of the main program. That's as direct as you can get.

So, when space is at a premium, or like many of us, you simply use the DOS itself, but would like to cut down on the command line typing, such as when assembling, running, debugging, and reassembling again. DIRECT is your choice. It's good to have a choice. I think the more you use it, DIRECT will be YOUR choice.

At $29.95, DIRECT, by Chris, can be ordered directly from Chris, at Microdex, 1212 N. Sawtelle, Tucson, AZ 85716 (602) 326-3502.

ATTENTION TRSDOS 1.3. USERS!

ANNOUNCING "SYSTEM 1.5.", THE MOST COMPREHENSIVE 1.3. UPGRADE EVER OFFERED!

MORE SPEED!! MORE POWER!! MORE PUNCH!!

While maintaining 100% compatibility to TRSDOS 1.3., this DOS upgrade advances TRSDOS 1.3. into the 90's!
SYSTEM 1.5. supports 16k-32k bank data storage and 4MGHZ clock speed (4/4P/4D).

DOUBLE SIDED DRIVES ARE NOW 100% UTILIZED! (all models).

CONFIG=Y/N	CREATES CONFIG BOOT UP FILE	DATE=Y/N	DATE BOOT UP PROMPT ON or OFF
TIME=Y/N	TIME BOOT UP PROMPT ON or OFF	CURSOR='XX'	DEFINE BOOT UP CURSOR CHAR
BLINK=Y/N	SET CURSOR BOOT UP DEFAULT	CAPS=Y/N	SET KEY CAPS BOOT UP DEFAULT
LINE='XX'	SET *PR LINES BOOT UP DEFAULT	WP=d.Y/N (WP)	WRITE PROTECT ANY or ALL DRIVES
ALIVE=Y/N	GRAPHIC MONITOR ON or OFF	TRACE=Y/N	TURN SP MONITOR ON or OFF
TRON=Y/N	ADD an IMPROVED TRON	MEMORY=Y/N	BASIC FREE MEMORY DISPLAY MONITOR
TYPE=B/H/Y/N	HIGH/BANK TYPE AHEAD ON or OFF	FAST	4 MGHZ SPEED (MODEL 4'S)
SLOW	2 MGHZ SPEED (MODEL III'S)	BASIC2	ENTER ROM BASIC (NON-DISK)
CPY (parm,parm)	COPY/LIST/CAT LDOS TYPE DISKS	SYSRES=H/B/'XX'	MOVE/SYS OVERLAY(s) TO HI/BANK MEM
SYSRES=Y/N	DISABLE/ENABLE SYSRES OPTION	MACRO	DEFINE ANY KEY TO MACRO
SPOOL=H/B.SIZE	SPOOL is HIGH or BANK MEMORY	SPOOL=D.SIZE='XX'	LINK MEM SPOOLING TO DISK FILE
SPOOL=N	TEMPORARILY DISABLE SPOOLER	SPOOL=Y	REACTIVATE DISABLED SPOOLER
SPOOL=RESET	RESET (NIL) SPOOL BUFFER	SPOOL=OPEN	OPENS, REACTIVATES DISK SPOOLING
SPOOL=CLOSE	CLOSES SPOOL DISK FILE	FILTER *PR.ADLF=Y/N	ADD LINE FEEDS BEFORE PRINTING 0DH
FILTER *PR.IGLF	IGNORES 'EXTRA' LINE FEEDS	FILTER *PR.HARD=Y/N	SEND 0CH to PRINTER (FASTEST TOF)
FILTER *PR.FILTER	ADDS 256 BYTE PRINTER FILTER	FILTER *PR.ORIG	TRANSLATE PRINTER BYTE TO CHNG
FILTER *PR.FIND	TRANSLATE PRINTER BYTE TO CHNG	FILTER *PR.RESET	RESET PRINTER FILTER TABLE
FILTER *PR.LINES	DEFINE NUMBER LINES PER PAGE	FILTER *PR.WIDTH	DEFINE PRINTER LINE WIDTH
FILTER *PR.TMARG	ADDS TOP MARGIN to PRINTOUTS	FILTER *PR.BMARG	ADDS BOTTOM MARGIN to PRINTOUT
FILTER *PR.PAGE	NUMBER PAGES, SET PAGE NUMBER	FILTER *PR.ROUTE	SETS PRINTER ROUTING ON or OFF
FILTER *PR.TOF	MOVES PAPER TO TOP OF FORM	FILTER *PR.NEWPG	SET DCB LINE COUNT TO 1
FILTER *KI.ECHO	ECHO KEYS to the PRINTER	FILTER *KI.MACRO	TURN MACRO KEYS ON or OFF
ATTRIB:d.PASSWORD	CHANGE MASTER PASSWORD	DEVICE	DISPLAYS CURRENT CONFIG INFO

All parms above are installed using the new LIBRARY command SYSTEM (parm,parm). Other new LIB options include DBSIDE (enables double sided drive by treating the "other side" as a new independent drive, drives 0-7 supported) and SWAP (swap drive code table #s). Dump (CONFIG) all current high and/or bank memory data/routines and other current config to a disk data file. If your type ahead is active, you can (optional) store text in the type buffer, which is saved. During a boot, the config file is loaded back into high/bank memory and interrupts are recognized. After executing any active auto command, any stored type ahead data will be output. FANTASTIC! Convert your QWERTY keyboard to a DVORAK! Route printer output to the screen or your RS-232. Macro any key,even F1, F2 or F3. Load *01-*15 overlay(s) into high/bank memory for a memory only DOS! Enter data faster with the 256 byte type ahead option. Run 4MGHZ error free as clock, disk I/O routines are properly corrected! Spool printing to high/bank memory. Link spooling to disk (spooling updates DCB upon entering storage). Install up to 4 different debugging monitors. Print MS-DOS text files, ignoring those unwanted line feeds. Copy, Lprint, List or CATalog DOSPLUS, LS-DOS, LDOS or TRSDOS 6.x.x. files and disks. Add top/bottom margins and/or page numbers to your hard copy. Rename/Redate disks. Use special printer codes eg: LPRINT CHR$(3); toggles printer output to the ROUTE device. Special keyboard codes add even more versatility. This upgrade improves date file stamping MM/DD/YY instead of just MM/YY. Adds optional verify on/off formatting, enables users to examine *01-*15, DIR, and BOOT sectors using DEBUG, and corrects all known TRSDOS 1.3. DOS errors. Upgrade includes LIBDVR, a /CMD driver that enables LIBRARY commands, such as DIR, COPY, DEBUG, FREE, PURGE, or even small /CMD programs to be used within a running Basic program, without variable or data loss.

By special arrangement with GRL Software,
SYSTEM 1.5. is now distributed exclusively by TRSTimes magazine.

ORDER YOUR COPY TODAY!

Send $39.95 (U.S. funds) to:

TRSTimes - SYSTEM 1.5.

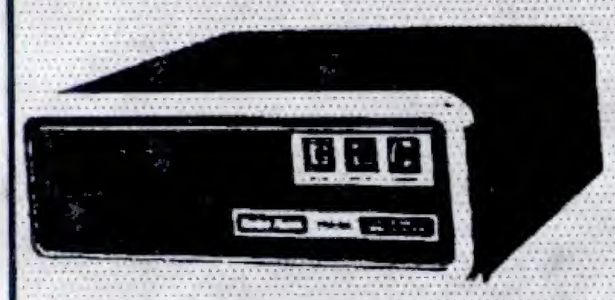
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367

HARD DRIVES FOR SALE

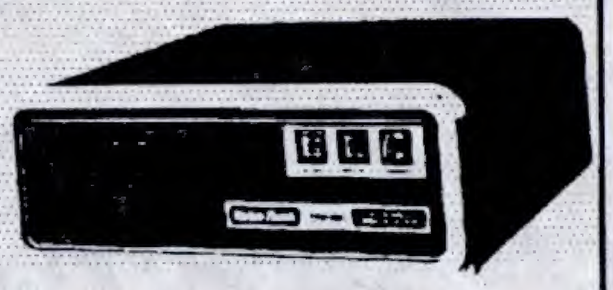
Genuine Radio Shack Drive Drive Boxes with Controller, Power Supply, and Cables.
Formatted for TRS 6.3, installation JCL included.
Hardware write protect operational.
Documentation and new copy of MISOSYS RSHARD5/6 included.
90 day warranty.

5 Meg $175 **10 Meg $225** **15 Meg $275** **35 Meg $445**

Shipping cost add to all prices



Roy T. Beck
2153 Cedarhurst Dr.
Los Angeles, CA 90027
(213) 664-5059



MISOSYS, Inc.

Aerocomp Hardware is now available from MISOSYS

Model I DDen Controller (DDC)	$45 + $6S&H
Model III/4 FDC board	$45 + $6S&H
Model III/4 RS232 board	$45 + $6S&H
Model III/4 RS232 Kit	$50 + $6S&H
Aerocomp 5 Meg HD	$250 + S&H
Aerocomp 20 Meg HD	$400 + S&H
Aerocomp 40 Meg HD	$500 + S&H
MM CP/M 2.2 HD drivers	$29.95 + $3S&H

***The MISOSYS Quarterly* subscriptions**

Keep up to date with the latest information on MISOSYS products, programs, patches, and articles in a professional magazine format. A subscription to *TMQ* will provide you with information, news, and announcements concerning our entire product line and related machine environments. **As a special for new subscribers, we'll start you off with issue VI.i and send you a complete past volume of four issues at no charge.** That's eight issues for the price of four! Subscriptions are: $25 US; $30 Canada; $35 Europe; $40 Australia

The Cornsoft Group
Model I/III Action Games Special

All five games on a single disk for one low price!
You get Bounceoids, Crazy Painter, Space Castle, Scarfman, and The Official Frogger; all five for $20 + $3S&H

PRO-WAM Year-End Price Plunge

If you are not using PRO-WAM on your 128K Model 4, you're not using your 4! You'll get a pop-up desktop manager with ADDRESS, BRINGUP, CAL, CARD, CALC, PHRASE, and more. Export|Import across windows. PSORT your data files. $37.48 + $6S&H

Sale prices good through Jan 31st.

MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170-0239
703-450-4181
orders: 800-MISOSYS

DoubleDuty doubles your 128K Model 4
Now on sale at half price!

DoubleDuty divides your Mod4's memory into three complete and independent partitions. Two operate as they were each their own 64K computer. Get the best task switcher you can buy. Our 2.6 release also works with extended memory. If you thought you needed a second computer, think again. At $24.98 + $3S&H, you can't afford to not have DoubleDuty.

GRAPHICS-90
for Model III TRSDOS 1.3/1.5 or LDOS

The definitive graphics program that Radio Shack SHOULD have released is now, through the courtesy of author Larry Payne and the hard work of Gary Shanafelt, available from TRSTimes. Our agreement is to sell this wonderful package at COST. Neither Larry Payne, Gary Shanafelt nor TRSTimes will make a dime on this venture. The price charged will cover the cost of reproducing the manual, disks, and mailing only. The only one profiting is YOU!!

GRAPHICS-90 $6.75

Shipping: U.S. & Canada: add $3.00
Europe: add $6.25 air mail or $4.25 surface mail
Asia, Austrailia & New Zealand: add $7.25 air mail or $4.25 surface mail
Please specify TRSDOS 1.3 or LDOS format

TRSTimes magazine - dept. GR90
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

RECREATIONAL & EDUCATIONAL COMPUTING

REC is the only publication devoted to the playful interaction of computers and 'mathemagic' - from digital delights to strange attractors, from special number classes to computer graphics and fractals. Edited and published by computer columnist and math professor Dr. Michael W. Ecker, REC features programs, challenges, puzzles, program teasers, art, editorial, humor, and much more, all laser printed. REC supports many computer brands as it has done since inception Jan. 1986. Back issues are available.

To subscribe for one year of 8 issues, send $27 US or $36 outside North America to REC, Att: Dr. M. Ecker, 909 Violet Terrace, Clarks Summit, PA 18411, USA or send $10 ($13 non-US) for 3 sample issues, creditable.

TRSTimes on DISK #9

Issue #9 of TRSTimes on DISK is now available, featuring the programs from the Jan/Feb, Mar/Apr and May/Jun 1992 issues:

TRSTimes on DISK is reasonably priced:

U.S. & Canada: $5.00 (U.S.)
Other countries: $7.00 (U.S.)

Send check or money order to:

TRSTimes on DISK
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

TRSTimes on DISK #1, 2, 3, 4, 5, 6, 7 & 8 are still available at the above prices